Fabio Roli
Sergio Vitulano (Eds.)

Image Analysis and Processing – ICIAP 2005

LNCS 3617

13th International Conference
Cagliari, Italy, September 2005
Proceedings

Springer

Lecture Notes in Computer Science 3617

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Fabio Roli Sergio Vitulano (Eds.)

Image Analysis and Processing – ICIAP 2005

13th International Conference
Cagliari, Italy, September 6-8, 2005
Proceedings

Springer

Volume Editors

Fabio Roli
University of Cagliari
Department of Electrical and Electronic Engineering
Piazza d'Armi, 09123 Cagliari, Italy
E-mail: roli@diee.unica.it

Sergio Vitulano
University of Cagliari
Department of Medical Sciences
Via S. Giorgio 12, 09124 Cagliari, Italy
E-mail: vitulano@pacs.unica.it

Library of Congress Control Number: 2005931520

CR Subject Classification (1998): I.4, I.5, I.3.5, I.2.10, I.2.6, F.2.2

ISSN 0302-9743
ISBN-10 3-540-28869-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28869-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11553595 06/3142 5 4 3 2 1 0

Preface

This volume contains the Proceedings of the 13th International Conference on Image Analysis and Processing (ICIAP 2005), held in Cagliari, Italy, at the conference centre "Centro della Cultura e dei Congressi", on September 6–8, 2005.

ICIAP 2005 was the thirteenth edition of a series of conferences organized every two years by the Italian group of researchers affiliated to the International Association for Pattern Recognition (GIRPR) with the aim to bring together researchers in image processing and pattern recognition from around the world. As for the previous editions, conference topics concerned the theory of image analysis and processing and its classical and Internet-driven applications.

The central theme of ICIAP 2005 was "Pattern Recognition in the Internet and Mobile Communications Era". The interest for such a theme was confirmed by the large number of papers dealing with it, the special session devoted to pattern recognition for computer network security, and the emphasis of two invited talks on Internet and mobile communication issues.

ICIAP 2005 received 217 paper submissions. Fifteen papers were collected into the two special sessions dealing with *Pattern Recognition for Computer Network Security* and *Computer Vision for Augmented Reality and Augmented Environments*.

The 143 papers selected by the Scientific Committee were organized into 10 oral sessions (38 papers), eight regular sessions and two special sessions, and three poster sessions (105 papers). In order to make the poster sessions the liveliest parts of the conference, this edition of ICIAP introduced a new type of presentation, namely "spotlight" presentations of the best papers selected for the poster sessions. Thirty-two poster papers out of 105 were presented in this form immediately before the related poster sessions.

The workshop program and this volume were enriched with the five invited talks, given by Horst Bunke (University of Bern, Switzerland), Giovanni Garibotto (Elsag, Italy), Anil K. Jain (Michigan State University, USA), Josef Kittler (University of Surrey, UK), and George Nagy (Rensselaer Polytechnic Institute, USA).

This edition of ICIAP also offered two tutorials, on *Computer Vision for Intelligent Vehicles* and *Biometric Systems*.

As for the previous editions, the best paper by a young author was selected by an international committee, and it was awarded the Caianiello Prize, in honor of Prof. Eduardo R. Caianiello, during the conference.

We wish to thank the Italian group of researchers affiliated to the International Association for Pattern Recognition (GIRPR) which gave us the possibility to organize this conference. We wish to express our appreciation to all those who helped to organize ICIAP 2005. First of all, we would like to thank all the

members of the Scientific Committee whose professionalism was instrumental in creating a very interesting technical programme. We also wish to thank Andrea Fusiello, Vittorio Murino, and Carlo Sansone who organized the two special sessions, Alberto Broggi, Raffaele Cappelli, and Giorgio Giacinto and Giorgio Fumera who served as tutorial and special session organizers.

It would have been impossible to organize the conference without the financial and technical support of the University of Cagliari and the Department of Electrical and Electronic Engineering, and both forms of support are gratefully acknowledged. We thank the International Association for Pattern Recognition and its Technical Committee TC1 on Statistical Pattern Recognition Techniques for sponsoring ICIAP 2005. We also wish to thank the Italian companies listed in the following pages for providing important financial support. Special thanks are due to the members of the Local Organizing Committee, Giorgio Giacinto, Roberto Perdisci, and Roberto Tronci, for their indispensable contributions to the Web site management, local organization, and proceedings preparation.

June 2005 Fabio Roli and Sergio Vitulano

Organization

General Chair

F. Roli (Univ. of Cagliari, Italy)

Program Chair

S. Vitulano (Univ. of Cagliari, Italy)

Steering Committee

V. Cantoni (Italy)
L. Cordella (Italy)
A. Del Bimbo (Italy)
G. Sanniti di Baja (Italy)
V. di Gesù (Italy)
M. Ferretti (Italy)
G. Vernazza (Italy)

Scientific Committee

M. Albanesi (Italy)
E. Ardizzone (Italy)
H. Bunke (Switzerland)
T. Caelli (Canada)
A. Campilho (Portugal)
L. Cinque (Italy)
R. Cipolla (UK)
C. Colombo (Italy)
R. Cucchiara (Italy)
L. De Floriani (Italy)
D. Doermann (USA)
B. Fisher (UK)
C. Garbay (France)
C. Guerra (Italy)
S. Impedovo (Italy)
A. Jain (USA)
J. Kittler (UK)
R. Klette (New Zealand)
M. Levine (Canada)
R. Molina (Spain)
V. Murino (Italy)
P. Mussio (Italy)
M. Petrou (UK)
V. Roberto (Italy)
H. Samet (USA)
A. Sanfeliu (Spain)
S. Tanimoto (USA)
G. Tascini (Italy)
J. Villanueva (Spain)
H. Yeshurun (Israel)
B. Zavidovique (France)

Local Organizing Committee

G. Giacinto (University of Cagliari, Italy)
R. Perdisci (University of Cagliari, Italy)
R. Tronci (University of Cagliari, Italy)

Tutorial and Special Session Committee

G. Giacinto (University of Cagliari, Italy)
G. Fumera (University of Cagliari, Italy)

Organized by

Dept. of Electrical and Electronic Engineering of the University of Cagliari
The Italian group of researchers affiliated to the IAPR (GIRPR)

Sponsored by

International Association for Pattern Recognition
IAPR Technical Committee TC1

Supported by

Ansaldo
Cap Research
Elsag
Regione Autonoma della Sardegna
Tiscali
Vitrociset

Additional Referees

Andrea Abate
Marco Aguzzi
Juan Andrade-Cetto
Stefano Berretti
Marco Bertini
Alberto Biancardi
Umberto Castellani
Dong Seon Cheng
Andrea Colombari
Luigi P. Cordella
Viguel Velhote Correia
Marco Cristani
Claudio De Stefano
Ciro D'Elia
Daniel DeMenthon
Haris Dindo
Michela Farenzena
Pasquale Foggia
Francesco Fontanella
Maria Frucci
Andrea Fusiello
Orazio Gambino
Costantino Gana
Gurman Singh Gill
Antoni Grau
Jean-Philippe Gravel
Adel Haffiane
May Huang
Francesco Isgrò
Stefan Jaeger
Samuel Kadoury
Marco La Cascia
Lionel Lacassagne
François Lecoat
Huiping Li
Jian Liang
Luca Lombardi
Ming Luo
Huanfeng Ma
Irene Macaluso
Alberto Machì
Paola Magillo
Angelo Marcelli
Roberto Marmo
Roberto Marzotto
Ana Maria Mendonça
Alain Merigot
Stefano Messelodi
David Mihalcik
Josep Maria Mirats
Alessandro Negrente
Michel Neuhaus
Francesca Odone
Laura Papaleo
Donovan Parks
Marcello Pelillo
Daniele Peri
Federico Pernici
Alfredo Petrosino
Roberto Pirrone
Andrea Prati
Roger Reynaud
Harkirat Sahambi
Jagan Sankaranarayanan
Jorge Scandaliaris
Andreas Schlapbach
Francesc Serratosa
Jorge Alves Silva
Nikom Suvonvorn
Dave Tahmoush
Francesco Tufano
François Verdier
Yingfeng Yu
Yang Yu
Yefeng Zheng
Gary Zi

Table of Contents

Invited Papers

Pattern Recognition for Computer Network Security

Computer Vision for Augmented Reality and Augmented Environments

Low and Middle Level Processing

Image Segmentation

Feature Extraction and Image Analysis

Graphs

Shape and Motion

Image Modelling and Computer Graphics

Image Communication, Coding and Security

Computer Architectures, Technologies and Tools

Multimedia Data Bases

Video Processing and Analysis

Pattern Classification and Learning

Stereo Vision

3D Vision

Medical Applications

Biometrics

Applications

Graph Matching – Challenges and Potential Solutions

Horst Bunke, Christophe Irniger, and Michel Neuhaus

Institute of Computer Science and Applied Mathematics,
University of Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
{bunke, irniger, mneuhaus}@iam.unibe.ch

Abstract. Structural pattern representations, especially graphs, have advantages over feature vectors. However, they also suffer from a number of disadvantages, for example, their high computational complexity. Moreover, we observe that in the field of statistical pattern recognition a number of powerful concepts emerged recently that have no equivalent counterpart in the domain of structural pattern recognition yet. Examples include multiple classifier systems and kernel methods. In this paper, we survey a number of recent developments that may be suitable to overcome some of the current limitations of graph based representations in pattern recognition.

Keywords: structural pattern recognition, graph matching, graph edit distance, automatic learning of cost functions, graph kernel methods, multiple classifier systems, graph database retrieval.

1 Introduction

The use of graph representations has become quite popular in pattern recognition. For an overview of recent literature, we refer the reader to a number of special issues that appeared on the subject [1,2,3,4] and the survey article [5]. Graph representations allow us to explicitly model relationships between different objects, or parts of the objects under consideration. This is a clear advantage over pattern representations based on feature vectors, which are restricted to using unary feature values. Moreover, in a graph representation one can use a variable number of entities, i.e. nodes and edges, depending on the complexity of the object under considerations. By contrast, in the statistical approach to pattern recognition, the number of features, i.e. the dimensionality of the feature space, needs to be defined a priori and is the same for all objects, no matter how simple or complex a particular object instance is.

On the other hand, there exist a number of problems with the use of structural pattern representation by means of graphs. First, we notice the high computational complexity of many operations on graphs. For example, computing the similarity of two graphs is typically exponential in the number of nodes of the two graphs involved, while the computation of the Euclidean distance of a pair of feature vectors needs only linear time. Secondly, the repository of algorithmic

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1–10, 2005.

procedures in the graph domain is quite limited when compared to the tools available for pattern representations through feature vectors.

From the general point of view, we notice that in the field of statistical pattern recognition, a number of powerful concepts emerged recently, which have no equivalent counterparts in the domain of structural pattern recognition yet. One example is multiple classifier systems [6], which have turned out to be a very powerful concept to improve current pattern recognition technology. Another example is kernel methods [7]. While a large number of kernel methods based on pattern representations in terms of feature vectors have been proposed, relatively little work has been published on kernels on graphs and symbolic data structures [8]. Consequently, researchers in structural pattern recognition are faced with a number of challenges, for example, to overcome the high computational complexity of graph matching, to take advantage of new developments in the fast growing field of multiple classifier systems, to make the power of kernel methods available in the graph domain, and more generally, to enlarge the repository of algorithmic tools for graph matching. In the current paper, we will address some of these challenges and sketch some potential solutions.

The rest of this paper is structured as follows. In the next section we introduce a novel approach for the automatic learning of edit cost functions. Then, in Section 3, kernel methods for structural pattern recognition are investigated, while Section 4 reviews recent work on multiple classifier systems using graph matching. Strategies for fast retrieval of graphs from large database are presented in Section 5, and conclusions are drawn in Section 6.

2 Automatic Learning of Edit Cost Functions

There are many tasks in structural pattern recognition, where one needs to measure the similarity, or distance, of a given pair of graphs. A suitable measure for this purpose is graph edit distance [9,10]. In its most general form, a graph edit operation is either a deletion, insertion, or substitution (i.e. a label change). Edit operations can be applied to nodes as well as to edges. By means of edit operations differences between two graphs are modelled. In order to enhance the modelling capabilities, often a cost is assigned to each edit operation. The costs are real numbers greater than or equal to zero. They are application dependent. Typically, the more likely a certain distortion is to occur the lower is its costs. The edit distance, $d(g_1, g_2)$, of two graphs is equal to the minimum cost taken over all sequences of edit operations that transform graph g_1 into g_2. Clearly, if $g_1 = g_2$ then no edit operation is needed and $d(g_1, g_2) = 0$. On the other hand, the more g_1 and g_2 differ from each other, the more edit operations are needed, and the larger is $d(g_1, g_2)$.

It is easy to see that the costs of the edit operations have a crucial impact on $d(g_1, g_2)$ [11]. Despite this impact, we notice a significant algorithmic deficit in structural pattern recognition because no procedures for the automatic learning of graph matching edit costs from a set of samples are available. While some potential solutions have been proposed for the case of string edit distance [12],

the edit costs in graph matching are still manually set in a heuristic trial and error procedure, exploiting problem specific knowledge. For examples see [13,14]. In this section we briefly outline a novel procedure for the automatic learning of the costs of graph edit operations from a set of sample graphs [15].

As one of the basic assumptions of the learning scheme described in the following, graphs with labels from the n-dimensional real space are considered. That is, each node label is a vector of fixed dimension consisting of n real numbers. Similarly, edge labels consist of a vector of m real numbers. The proposed scheme takes a sample set of graphs as input and tries to minimize the average edit distance between each pair of graphs from the same class by suitably adjusting the costs of the underlying edit operations. This is equivalent to minimizing the average intra-class distance of a given set of graphs. The proposed scheme is based on self-organizing map, SOM [16]. There is one SOM for each type of edit operation, i.e. node deletion, node insertion, node substitution, edge deletion, edge insertion, and edge substitution. For example, the map for node substitution is a n-dimensional grid representing the space of node labels (i.e. n-dimensional real vectors). The cost of a node label substitution is proportional to the Euclidean distance between the two corresponding locations in the grid. The SOM learning procedure starts with a non-deformed n-dimensional grid. It computes the edit distance of a pair of graphs and moves each pair of grid points that correspond to a substitution closer to each other. In this way, the Euclidean distance of a pair of labels that are often substituted one by another, is iteratively minimized, which leads to a smaller overall graph edit distance between the two involved graphs. For more details we refer to [15].

The proposed learning scheme has been successfully applied in a recognition experiment involving synthetically generated characters. In addition the SOM-based learning method has been used for the identification of diatoms. Here graphs derived from real images of diatoms were involved and with the automatically derived costs a higher recognition accuracy than with manually chosen costs was achieved. Another procedure for automatically learning the costs of graph edit operations, based on the EM algorithm, has been proposed in [17].

3 Graph Matching Using Kernel Methods

Kernel methods have seen an increasing amount of interest in the pattern recognition community in recent years [7]. On one hand, the theory of kernel methods is well studied and provides us with a number of convenient theoretical results [18,19]. On the other hand, kernel machines have been successful in outperforming other types of classifiers on standard datasets [20]. Even more relevant to the present paper is that they also allow us to extend the repository of algorithms for structural pattern recognition in a very elegant way: once suitable kernel functions have been designed, a large number of methods for pattern classification, clustering, dimensionality reduction etc. become directly applicable to graphs.

With the rise of kernel machines in pattern recognition, a number of structural kernels for strings and graphs have been developed. A kernel for text categorization, for instance, has been developed that encodes the number of occurrences of words in texts [21]. In another approach, a kernel on string and graph data has been proposed that basically describes structures by means of the substructures they contain [22,23]. Another class of kernel functions defined on graphs are marginalized kernels [8,24], which are derived from random walks on attributed graphs. In [25] the authors propose a graph matching system for unconstrained large graphs based on functional interpolation theory. These methods can be characterized by explicitly addressing the matching problem via a kernel function. In this section we describe an alternative approach, originally proposed in [26], where we leave the structural matching to the error-tolerant edit distance algorithm and use kernel machines to subsequently carry out the classification task.

The kernel function introduced in [26] is defined as follows:

$$K(g_1, g_2) = \frac{1}{2}\left(d(g_1, g_0)^2 + d(g_0, g_2)^2 - d(g_1, g_2)^2\right),$$

where $d(g, g')$ is the edit distance between any two given graphs, g and g', and g_0 is a (reference) graph that needs to be defined by the system designer. We observe that this kernel computes the difference of two distances, the first being the squared distance of g_1 to g_0 plus the squared distance of g_0 to g_2, while the second is the direct squared distance of graphs g_1 and g_2.

In general it is not guaranteed that function $K(g_1, g_2)$ is a valid kernel. However, it has a number of interesting properties that justify its use as a kernel function. First, the distance of two vectors in the dot product space is equal to the edit distance of the corresponding graphs. Secondly, element g_0 plays the role of a null, or zero, element in the dot product space. The length of its corresponding vector is zero and the angle it forms with any other element in the dot product space is undefined. For a detailed analysis and a derivation of these properties see [26]. Given the kernel function defined above, one can build more complex kernels by selecting more than one zero graphs and combining the resulting kernels, for example, by their sum or product.

An exhaustive experimental evaluation of the proposed kernel method on eight different data sets has been reported in [26], including not only graph, but also string representations of patterns. In this experimental evaluation a support vector machine (SVM) using the proposed kernel was compared to a nearest neighbor classifier that works directly on the edit distances computed in the graph (or string) domain. This experimental evaluation has shown that the kernel based SVM very clearly outperforms the nearest neighbor classifier in the edit distance space.

4 Multiple Classifier Systems Using Graph Matching

Multiple classifier systems have become a very active field of research in pattern recognition [6]. The basic idea is to combine the output of several different classi-

fiers in order to build one large, and hopefully more accurate, multiple classifier system. From the general point of view, we can distinguish between two principal approaches to building a multiple classifier system. The first approach consists in developing each of the classifiers to be included in the system individually, i.e. the classifiers of the ensemble are built independently. Secondly, methods such as bagging, boosting, or random subspace are applied to one base classifier in order to generate the ensemble members automatically [27,28,29]. In this case only the base classifier needs to be constructed manually. An important issue in multiple classifier systems is diversity. Only if the individual members of the ensemble are diverse, i.e. independent of each other in the ideal case, one can expect that a multiple classifier system is able to outperform the best of its individual members.

Almost all work in the domain of multiple classifier systems has focussed on statistical, i.e. feature vector based, classifiers. However, because structural classifiers are usually very different from statistical ones, one can expect a substantial gain in diversity by including structural classifiers in an ensemble, i.e. by mixing statistical and structural classifiers in a multiple classifier system. Although this approach seems very appealing, little work of this kind has been reported [30,31]. Even more striking is the observation that almost no attempts are known from the literature to create ensembles of structural classifiers automatically from one given base classifier. In this section we first show how the performance of an image classification and retrieval system can be enhanced by mixing graph matching and feature vector based classifiers with each other [32]. Next a system for text classification based on an automatically generated ensemble of structural classifiers is described [33].

Image classification is an important task in image database retrieval and similar applications. Often binary classifiers are applied that have to decide, given an unknown image, whether or not it belongs to a certain predefined category, for example *people*, *countryside*, *city* etc. A number of classifiers have been proposed for that purpose. In [34], for example, a SVM using the values of the image histogram as features has been described. Such a classifier can be expected to perform well on cases where the salient features of an image category are of global nature and the discrimination of classes does not depend on local image regions. Also in [35] a SVM was used for the purpose of image classification. However, in contrast with [34], the presence or absence of particular types of regions is used as basic SVM features. The salient regions are automatically learned from a training set. This classifier is particularly meaningful when some region types are highly discriminative. For example, skin region types are a strong hint to images with people. However, this classifier certainly suffers if image segmentation is poor. As a third classifier, a graph matching based nearest neighbor classifier was proposed in [14]. Images are represented as region adjacency graphs, where region properties are used as node attributes, and properties such as the distance or the length of the common boundary between two adjacent regions serve as edge attributes. The graph edit distance between an unseen sample and each element of the training set is computed and the sample is assigned to its nearest

neighbor in the training set. In contrast with the other two classifiers, the graph based method takes spatial relationships and distance between pairs of regions into account.

Obviously, the three individual classifiers discussed in the previous paragraph are quite diverse. Hence it seems meaningful to combine then into a multiple classifier system. In order to accomplish such a combination, one needs to transform the individual classifiers' outputs into a common format. In [32] such a transformation is described. The resulting multiple classifier system was able to significantly outperform the best individual classifiers in a number of experiments under various combination rules.

Due to the explosion of material on the internet, the automatic classification of text has become a very important area of research. In a recent paper the use of graphs, rather than the standard vector model, has been proposed as a formal model for document representation [36]. The maximum common subgraph based distance measure introduced in [37] and a simple nearest neighbor classifier have been used in order to assign predefined text categories, such as *business*, *computer*, *sports*, *politics* etc. to previously unseen text documents. In an experimental evaluation it has been shown that this approach is able to outperform classifiers that use the traditional vector model of document representation.

In [33] this approach has been enhanced by automatically generating a multiple classifier system out of the base classifier used in [36]. The basic idea is to randomly select subsets of node labels (i.e. words from the underlying dictionary) an take, in the prototypes used by the nearest neighbor classifier, only those nodes into account that are labelled with an element from the chosen subset. This is in analogy to the random feature subset selection method proposed in [29] for the case of classifiers in a feature space. In an experimental evaluation it was shown that the resulting classifier ensemble, which has been automatically generated, is able to outperform the original graph matching based classifier.

5 Fast Retrieval of Graphs from Large Databases

It is well known that graph matching in general is a computationally expensive procedure. In many applications, particularly in pattern recognition and information retrieval, the computational complexity of graph matching is further increased by the need to match an input graph against an entire database of graphs. A variety of mechanisms have been proposed to reduce this additional factor [38,39,40,41]. However, they are generally unsuitable for processing large databases of graphs.

Recently, graph database retrieval has been addressed using machine learning techniques [42,43,44]. The idea is to preprocess the graph database extracting a feature vector representation of the graphs. Following feature vector extraction the database is then "data mined" using machine learning, specifically decision tree, techniques. At runtime, based on an input sample and a given matching paradigm (for example, graph isomorphism, subgraph isomorphism, or error-tolerant matching), all valid candidates are retrieved from the database

by traversing the decision tree previously induced. This approach is referred to as filtering the graph database. In filtering, the size of a database is reduced by first ruling out as many graphs as possible using a few simple and fast tests. After the filtering phase, an ordinary exact matching algorithm is applied to the remaining database graphs.

The matching paradigms considered in [42,43] include graph as well as subgraph isomorphism (both from the input graph to the database and from the database graphs to the input). For graph isomorphism, a necessary condition for two graphs g_s and g_{db} being isomorphic is that they have identical feature values. Given a sample graph g_s, the decision tree induced during preprocessing can be used to compare the most significant features identified on the database, ruling out a great number of graphs to be tested by a full-fledged isomorphism algorithm. The approach used for graph isomorphism filtering can be extended to subgraph isomorphism filtering in a straightforward way. The basic idea is that the feature values of the subgraph g_s occur at most as many times as they do in the designated supergraph g. For this matching paradigm, there exist two filtering methods. Decision trees induced for graph isomorphism filtering can be used to identify subgraph isomorphism candidates if the traversal algorithm is modified [43]. The other approach is to alter the decision tree structure, in which case the same traversal algorithm can be used for both tree types, graph isomorphism as well as subgraph isomorphism trees [42].

In [44], an extension of decision tree filtering to error-tolerant matching, i.e. retrieval of graphs with an edit distance to the input that is smaller than some given threshold distance, is presented. The approach is based on the concept of imposing a lower limit on the size of a possible maximum common subgraph between database graphs and input sample. According to the feature vector representation of the graphs an estimate on the size of the maximum common subgraph between input sample and graphs in the database is derived. If the estimated size is below the given input threshold it is certain that the distance between the graphs is larger than required and the graphs can be ruled out from the candidate set. In a number of experiments, this algorithm has been found to be very effective for database filtering [44].

6 Conclusions

Most of the algorithms commonly used in pattern recognition and intelligent data analysis are based on object representations in terms of feature vectors. This representation formalism is advantageous in the sense that a rich repository of algorithmic tools is available, including subspace projection techniques, methods for clustering and classification, and others. On the other hand, feature vectors are limited because they can represent only unary properties and usually assume that the same number of attribute values being measured on each object.

Symbolic data structures, including strings, trees, and graphs, are very suitable to overcome these limitations. However using such kind of object representation, we are facing the problem of increased computational complexity and

the lack of suitable mathematical tools for a number of tasks. To overcome these problems is a clear challenge to the community of researchers in pattern recognition.

Recent work has resulted in a number of promising approaches, making the repository of algorithmic tools for graph matching richer, for example, through procedures for the automatic learning of edit cost functions from sample sets of graphs. Work on graph kernels and graph matching based kernel functions is also of crucial importance for this purpose. With the availability of suitable kernels we may expect that a large class of powerful methods for intelligent data analysis, originally developed for feature representations, become instantly available in the graph domain. However, there is still a long way to go until mature kernel based methods for graphs and other structural data representations are at our disposal. Another challenging domain is multiple classifier system. While good progress in the field can be observed as far as statistical pattern recognition is concerned, the use of structural classifiers and their automatic generation in the context of multiple classifier systems is still in its infancy. We argue that also the problem of efficient graph retrieval from large databases is a hard problem, providing still a number of challenges.

Acknowledgment

The authors would like to thank Drs. Bertrand Le Saux, Adam Schenker, Mark Last, and Abe Kandel for fruitful cooperation and contributions to this paper. This research was supperted by the Swiss National Science Foundation NCCR program "Interactive Multimodal Information Management (IM)2" in the Individual Project "Multimedia Information Access and Content Protection" as well as the Project "Matching and Retrieval of Graphs from Large Graph Databases" (Nr. 2100-066700) of the Swiss National Science Foundation. The authors thank the Foundation for the support.

References

1. IEEE Transactions on Pattern Analysis and Machine Intelligence: Special section on graph algorithms and computer vision. **23** (2001) 1040–1151
2. Pattern Recognition Letters: Special issue on graph based representations. **24** (2003) 1033–1122
3. Int. Journal of Pattern Recognition and Art. Intelligence: Special issue on graph matching in pattern recognition and computer vision. **18** (2004) 261–517
4. Special Section on Syntactic and Structural Pattern Recognition: IEEE Transactions on Pattern Analysis and Machine Intelligence. **27** (2005) to appear
5. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Int. Journal of Pattern Recognition and Artificial Intelligence **18** (2004) 265–298
6. Roli, F., Kittler, J., Windeatt, T., eds.: Proc. 5th International Workshop on Multiple Classifier Systems MCS. LNCS 3077, Springer (2004)

7. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
8. Gärtner, T.: A survey of kernels for structured data. ACM SIGKDD Explorations **5** (2003) 49–58
9. Bunke, H., Allermann, C.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters **1** (1983) 245–253
10. Sanfeliu, A., Fu, K.: A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, Man, and Cybernetics **13** (1983) 353–363
11. Bunke, H.: Error correcting graph matching: On the influence of the underlying cost function. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 917–922
12. Ristad, E., Yianilos, P.: Learning string-edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 522–532
13. Ambauen, R., Fischer, S., Bunke, H.: Graph edit distance with node splitting and merging, and its application to diatom identification. In Hancock, E., Vento, M., eds.: Proc. 4th IAPR Int. Workshop on Graph Based Representations in Pattern Recognition. LNCS 2726, Springer (2003) 95–106
14. Le Saux, B., Bunke, H.: Feature selection for graph-based image classifiers. In: Proc. 2nd Iberian Conf. on Pattern Recognition and Image Analysis IbPRIA. LNCS, Springer (2005) to appear
15. Neuhaus, M., Bunke, H.: Self-organizing maps for learning the edit costs in graph matching. IEEE Transactions on Systems, Man, and Cybernetics **35** (2005) to appear
16. Kohonen, T.: Self-organizing Maps. Springer Verlag (1995)
17. Neuhaus, M., Bunke, H.: A probabilistic approach to learning costs for graph edit distance. In Kittler, J., Petrou, M., Nixon, M., eds.: Proc. 17th Int. Conference on Pattern Recognition. Volume 3. (2004) 389–393
18. Vapnik, V.: Statistical Learning Theory. John Wiley (1998)
19. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press (2002)
20. Byun, H., Lee, S.: A survey on pattern recognition applications of support vector machines. Int. Journal of Pattern Recognition and Artificial Intelligence **17** (2003) 459–486
21. Joachims, T.: Learning to Classify Text Using Support Vector Machines. Kluwer Academic (2002)
22. Watkins, C.: Dynamic alignment kernels. In Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D., eds.: Advances in Large Margin Classifiers. MIT Press (2000) 39–50
23. Lodhi, H., Saunders, C., Shawe-Tayler, J., Cristianini, N., Watkins, C.: Text classification using string kernels. Machine Learning **2** (2002) 419–444
24. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Proc. 20th Int. Conf. on Machine Learning. (2003) 321–328
25. van Wyk, M.A., Durrani, T.S., van Wyk, B.J.: A RKHS interpolator-based graph matching algorithm. PAMI **24** (2002) 988–995
26. Neuhaus, M., Bunke, H.: Edit distance based kernel functions for structural pattern classification (2005) submitted.
27. Dietterich, T.: Ensemble methods in machine learning. In: Proc. 1st Int. Workshop on Multiple Classifier Systems. LNCS, Springer (2000) 1–15
28. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
29. Ho, T.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 832–844

30. Marcialis, G., Roli, F., Serrau, A.: Fusion of statistical and structural fingerprint classifiers. In: Proc. 4th International Conf. Audio- and Video-Based Biometrie Person Authentication AVBPA. (2003) 310–317
31. Serrau, A., Marcialis, G., Bunke, H., Roli, F.: An experimental comparison of fingerprint classification methods using graphs. In Brun, L., Vento, M., eds.: Proc. 5th IAPR Int. Workshop on Graph Based Representations in Pattern Recognition. LNCS 3434, Springer (2005) 281–290
32. Le Saux, B., Bunke, H.: Bayesian multiple classifier system for image content recognition (2005) submitted.
33. Schenker, A., Bunke, H., Last, M., Kandel, A.: Building graph-based classifier ensembles by random node selection. In Roli, F., Kittler, J., Windeatt, T., eds.: Proc. 5th Int. Workshop Multiple Classifier Systems MCS. LNCS 3077, Springer (2004) 214–222
34. Chapelle, O., Haffner, P., Vapnik, V.: Svms for histogram-based image classification. IEEE Transactions on Neural Networks **10** (1999) 1055–1065
35. Le Saux, B., Amato, G.: Image recognition for digital libraries. In: Proc. 6th ACM MultiMedia/Int. Workshop on Multimedia Information Retrieval. (2004) 91–98
36. Schenker, A., Last, M., Bunke, H., Kandel, A.: Classification of web documents using graph matching. Int. Journal of Pattern Recognition and Artificial Intelligence **18** (2004) 475–496
37. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters **19** (1998) 255–259
38. Messmer, B., Bunke, H.: A new algorithm for error–tolerant subgraph isomorphism detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 493–505
39. Shapiro, L., Haralick, R.: Organization of relational models for scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **4** (1982) 595–602
40. Lopresti, D., Wilfong, G.: A fast technique for comparing graph representations with applications to performance evaluation. Int. Journal on Document Analysis and Recognition **6** (2004) 219–229
41. Giugno, R., Shasha, D.: Graphgrep: A fast and universal method for querying graphs. In: Proc. 16th Int. Conference on Pattern Recognition. Volume 2. (2002) 467–470
42. Irniger, C., Bunke, H.: Decision tree structures for graph database filtering. In Fred, A., ed.: Structural, Syntactic, and Statistical Pattern Recognition, Proc. Joint IAPR Int. Workshops SSPR and SPR. LNCS 3138, Springer (2004) 66–75
43. Irniger, C., Bunke, H.: Graph database filtering using decision trees. In Kittler, J., Petrou, M., Nixon, M., eds.: Proc. 17th Int. Conference on Pattern Recognition. Volume 3. (2004) 383–388
44. Irniger, C., Bunke, H.: Decision trees for error-tolerant graph database filtering. In Brun, L., Vento, M., eds.: Proc. 5th IAPR Int. Workshop on Graph Based Representations in Pattern Recognition. LNCS 3434, Springer (2005) 301–312

How to Make Business with Computer Vision Technology

Giovanni B. Garibotto

Elsag spa, Genova, Italy
giovanni.garibotto@elsag.it

Abstract. Business development in highly competitive new markets is strongly dependent on the level of technology innovation. Actually, the real success of a new product is affected by many other factors in the development chain, including customer requirements understanding and system implementation. The paper is aimed to refer some direct experience of New product development based on Computer Vision Technology. A practical example is referred to describe a recent success story in the field of Security application, a mobile Automatic Number Plate Recognition system that is currently used by many security forces in different countries in Europe and the USA.

1 Business Development by Technology Innovation

There is a widespread agreement that technology innovation is a key to economic growth. Actually social and political forces determine the context within which R&D technology investments occur and strongly influence perceived rewards, risks and tradeoffs. Industry-market trends and factors determine opportunities due to market maturity, number of competitors, etc. Rouse [1] concludes that company's abilities to recognize changing relationships with evolving markets are key factors to continued success. The time until returns is usually quite long. Academic research often requires 20 years until economic gain [2]. Actually much shorter time horizons are usually considered as useful, especially in the industry where most application programs are often limited to yearly budgets.

Moreover, R&D technology efforts tend to achieve the greatest benefits in unanticipated domains and markets [1]. A key issue is the identification of the role of R&D technology in the overall organization.

Many managers in the past saw the R&D function as a knowledge incubator, where a wide range of ideas are pursued and various invention developed. A much more agreed answer today is considering R&D function to provide enabling technologies to support market penetration. Moreover, it should give key competitive advantages by creating leading-edge products and processes. Thus, R&D objectives are more directly linked to business objectives than in the past.

Another important question is related to the capability to measure success of R&D-technology innovation. Traditional metrics based on the number of publications and patents are no more sufficient. Much more emphasis is given on measurable value added solutions of technical and operational problems (product and process innovation).

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 11 – 18, 2005.

1.1 Technology Transfer Actions

At the international level many institutions are already promoting technology transfer from basic research to product development. One example is given by the IST innovation prize that has been organized by EEC [3] to promote innovation and technology transfer activities, from the University to the industry and to the market. Traditional university structures have not been designed for technology transfer purposes but for research and education only. As a result, there are often organization constraints that limit universities' abilities to carry out technology transfer efficiently with industrial companies or to commercialize the results of their research. A few relevant changes are required in the academic context to fulfill pre-requisites for efficient and successful collaborations with industry. Main challenges are to move from a given budget to more competitive markets, from scientific curiosity to concrete market needs, from continuity to improved flexibility, from bureaucracy to management efficiency, from the ivory tower of science to international competition.

The most common model of technology transfer process is the linear model that has been used as a grounding theory for all science parks and incubators. According to the linear model the process starts with basic scientific research and passes through applied and more developmental research activities, to achieve the identification of new products and new process ideas, the evolution and testing of prototypes, to get commercial production and finally to achieve and consolidate an effective market position. The linear model is quite appropriate to highlight the relationship between long-term scientific research and its commercialization and continues to dominate much science and technology policy-making [4]. Anyway, a few main critical issues have been found to linear models. Actually, research results and ideas are used at all stages of the innovation process and not only in the early phase and the relationship between basic research and commercialization is too complex to be described as a linear function and requires a series of feedback loops and changing actions to succeed. Finally, the linear models tend to underestimate the contributions of people involved in the innovation process, including users whose ideas are often another relevant starting point for innovation.

1.2 Evaluating a Technology Transfer Project

It is important to evaluate technology transfer programs and projects on an ongoing basis to determine their success and to incorporate lessons learned in future activities. Recent studies of just over 400 examples of the application of technology from the National Aeronautics and Space Administration (NASA) demonstrated that the technology users benefited from sales or savings of $33.5 billion because of successful technology transfer. In a similar study for the Agricultural Research Service (ARS), sales or savings totaling $14.3 billion were identified in 87 cases where users applied ARS technology or science. Technology transfer is rarely accomplished in a short period of time. The process may not be complete for five or ten years. Therefore, it is not always possible to wait for the end results; it is sometimes necessary to seek interim indicators that will demonstrate progress toward the particular project goal. It is important to identify intermediate products whenever possible. For example, a project that uses remediation technology might include plans

to use a variety of workshops and demonstrations as interim steps. Some experts say a transfer is successful only when it becomes a profitable product or process, while others claim a transfer is successful when the technology is at least reviewed for possible use by another person or organization.

2 Computer Vision Market

Different areas of markets are currently addressed by Image Processing and Computer Vision technology. A very simple and basic main classification of the different market sectors can be arranged as follows:

- Image quality inspection for industrial applications (industrial automation, pharmaceutical and manufacturing, etc.) represents a mature market field with many specialized Exhibition and Congress.
- OCR (Optical Character Recognition) in postal automation and document processing represents still one of the most important business success of Image Processing and Recognition technologies.
- Biomedical Image Processing for Diagnostic systems and surgical systems is an essential component of all new advanced devices in the medical field.
- Robotics, Automation in different areas (Space, applications, service robotics and Logistic automation)
- Image Processing in Telecommunications and Multimedia represents another relevant component of technology innovation
- Security and Surveillance Applications is the emerging area with particular interest in video-surveillance, face-recognition and biometric control, human motion analysis and behavior control.

3 A Success Story: Auto-Detector

Auto-Detector [5] is a new mobile Automatic Number Plate Recognition system, installed on board of any kind of patrol vehicle (car, motor-vehicle), to automatically detect and read the license plates of the vehicles falling in its field of view. The innovation content of Auto-Detector is twofold: it is a new service solution for security and surveillance monitoring; moreover it represents a significant innovation of integrated Computer Vision solutions by continuous high-rate number plate recognition from a moving sensor in all possible environmental conditions.

Auto-Detector is an independent intelligent sensor that is continuously inspecting what happens around the patrol vehicle and is able to detect automatically the presence of a license plate irrespective of its orientation in the field of view. As such the proposed system represents a revolutionary contribution to patrol crew, working in background and without affecting normal patrol duties.

The continuous recognition of plates in the scene is a performance far exceeding any practical possibility by the human eye and the on-line response feature of Auto-Detector (by on-board real-time checking the recognized plate against a search list size of more than millions of plates) provide a great economical value for all security and surveillance applications. All recognized license plates are immediately compared

on board, against one or more lists of selected number-plates according to different levels of investigation priorities. They may be either alarm hot-list (including plates under investigation, as stolen cars or searched vehicles) as well white list of authorized vehicles (access control to downtown restricted areas).

3.1 The Industrial Context

Elsag spa is a benchmark player in the Italian Information & Communication Technology market, providing services and solutions in pursuit of its mission to "*Satisfy the requirements of medium and large companies by adopting innovative technologies to design, implement and manage ICT services and solutions*". ELSAG Company has been established in 1905. It is a medium-large size company with approximately 2950 employees (2003). Elsag belongs to Finmeccanica spa with leadership in Information Technology, in the fields of Automation, Information and Physical Security and Defense. In the last few years there has been a significant development of new surveillance applications in the traffic control market, with an increasing connection to institutional security customers (Ministry of the Defense and Ministry of the Interior and police departments). Elsag is qualified as a high level system integration of complex and high performance systems, with Information Technology components. Because of this system integration approach, the development of new products is always considered as an added value component of the Elsag's offer in the complete integrated system.

3.2 Innovation Content of Computer Vision and Image Processing

The main contribution of technology innovation in the Auto-Detector project is provided in the Computer Vision process and Optical Character Recognition to achieve a detection performance better than 90% of all license plates in the field of view and a correct recognition rate greater than 99% among all detected license plates. Moreover the system has been equipped also with suitable and effective learning tools, to achieve successful performance for all international number plates in all countries. One of the key features of the Auto-Detector is the high-performance Elsag O^2CR technology that has been strongly improved to read License Plates continuously from the video stream at the highest rate (more than 25 frame per second fps, for progressive cameras) without any need of external triggering devices. This technology is based on Elsag's long experience in Intelligent Character Recognition applied to document and form processing as well as for postal address understanding.

Adaptive exposure control is an essential feature of the acquisition system. Computer Vision and Image Processing technologies are used to achieve the most effective image normalization and optimize character recognition (in a mobile application License Plates may appear with a perspective 3D deformation greater than 30°). The recognition process consists in three main blocks: Plate detection, Recognition Measure and Contextual Analysis (exploiting both spatial and syntactic information to select the best hypothesis). Temporal post-processing performs data fusion and tracking among different images of the video sequence.

Another important innovation contribution is the miniaturization of the imaging sensor case, to be installed on-board within a very small space (lighting bar, roof of

the car, interior, etc.). To achieve such goals the proposed solution is an effective optimization of existing technology components in the area of digital cameras (LVDS and Camera-Link standard) and infrared illumination sources (using a very innovative LED-on-chip technology that was originally developed and used for automation inspection and Machine Vision). Finally, the selected Auto-Detector on-board processing unit is an effective network processor using a local LAN connection between dedicated processors devoted to each camera and data storage and search, with low-power consumption (less than 15 W), and automotive constraints.

3.3 Main System Features

The use of a mobile License Plate Reader (LPR) system provides some important advantages in security and ITS applications. Data collection is no more limited to a fixed position along the road (like a tripod or a fixed pole), but can be performed almost everywhere within the traffic flow. Auto-Detector is a fully integrated system comprising the on-board Computer Vision and ANPR recognition subsystem, and a central supervisory station for data collection (upload-download) as well as data search and update and plate list management. The proposed mobile Auto-Detector system fits exactly the expectations and requirements of the customers.

The on-board subsystem integrates ANPR recognition functions with an effective and ergonomic man-machine interface including other service functions of communication (public or private radio link) and geo-reference (through GPS and local encoder), with optional interface to standard navigator subsystems.

On-board man-machine interface is also used to manage online communication with the control centre, to display license plates recognised during patrols and to provide alarm messages (blacklisted licence plate detected). A key issue of Auto-Detector is camera sensor configuration to manage unpredictable variations in lighting conditions, from just a few lux in areas of shadow, tunnels and night operations, to more than 100 Klux in full sunlight. The integrated acquisition system comprises a digital micro-camera coupled to a programmable high-speed pulsed LED illuminator in the near infrared spectrum to limit ambient light interference.

An important feature of Auto-Detector is its extremely easy operation that greatly improves user acceptance. When starting their mission, patrol officers are not required to perform any supplementary operations. All data-update operations are performed completely automatically over the wireless-LAN link. During patrols, officers receive and send messages from/to the Operations Centre through the available interface monitor. When a license plate in the hot-list is recognised, an alarm signal starts and the acquired digital images are immediately displayed on screen to provide the human operator evidence of the positive detection. The alarm signal is automatically sent also to the Operations Centre through the onboard communication system in real time. A marker is automatically entered on the onboard navigator map to display the position (on the navigation map) where the license plate was detected and recognised. On returning from patrols, the system automatically uploads the data it has collected to the central unit over the wireless-LAN link and then shuts the system down. The storage of vehicle transit data and images is managed according to the privacy rules established in the different countries.

Fig. 1. Auto-Detector has been installed on-board of police patrol cars, since 2002

3.4 Brief History of the Project

Elsag has developed its first ANPR system at the beginning of the 90's for traffic control applications, from pay-toll highway access control (Telepass by Italian Autostrade), parking control, access control to restricted traffic areas.

The mobile Auto-Detector project started in 2001 as a technology-push by the Technology Research department on Imaging and Computer Vision with a first prototype to demonstrate feasibility of the mobile recognition function. Such early experience has pinpointed a few main problems dealing with reading performance, computational speed, sensor miniaturization, automotive requirements for the processing unit (compact and low-power consumption). At the same time the Security Institutions in Italy (Police and Carabinieri Army) were looking for advanced solutions able to read license plates on-board of patrol cars, to support the heavy (almost impossible) task of patrol crews to search hot-list vehicles in the normal traffic flow.

On March 2002 a first demonstration to the Police department has been issued. This initiative proved the great potential of the proposed system and provided some important feedback in terms of application requirements, ergonomic issues and system constraints (communication and operational procedures). The product has been protected by an international patent "Surveillance System for Homeland security through Mobile Automatic Number Plate Recognition of Vehicles in transit"; by G. Garibotto, P. Castello, E. Del Ninno, submitted to the PCT (Patent Cooperation Treaty) through all main countries (Europe, USA, Canada, South-America).

On July 2003 a demo-kit of the system was realized. It consists of two small cases. The first case contains the processing unit to be placed in the baggage compartment of the car (with power connection to the light-power 12V DC of the car). The second case contains cables and the sensor box, with magnetic support, so that it can be easily placed on top of any existing car. Man-machine interface is granted by a portable laptop PC with wireless-LAN Wi-Fi connection. In 2004 International applications

started with a series of demonstrations that have beeen arranged in the international market., with participation to the most qualified international fairs. On August 2004 a joint company has been founded between Elsag and Remington Arms (US) whose name is RELES (Remington-Elsag Law Enforcement Systems) [6] to commercialize the Auto-Detector system in the US market. Since the beginning of 2005 the US market has been addressed with extremely positive results [Ohio]; moreover, additional applications have been launched in other countries (South Africa, Turkey, China, etc.). The overall direct cost of the development project has been covered by internal funding of the company but the basic competence and technology has been achieved through the participation to different national and international projects. A relevant contribution came from the Italian Ministry of Education, University and Research (MIUR) in a project led by Elsag, entitled "Neural systems for service and plant automation", corresponding to Theme 5 of the "National Program for Microelectronics and Bioelectronics" (2000-2003). Basic knowledge on Computer Vision and Image processing Technology has been developed within the European projects of ESPRIT P940 "Depth and Motion Analysis" (86-92) and more recently within the project VISOR IST –1999-10808 (2000-2002), devoted to video monitoring applications. Further investments are in progress by Elsag to support technology and product evolution, to improve recognition performance, increase processing speed, improve resolution and range of operation as well as to introduce additional services and functions to patrol car operators.

3.5 Related Products on the Market

At the very beginning of the project (2002) the other competitive systems were definitely less performing and quite limited (in speed and resolution). During the last year (2005) the number of competitors in the international market has been steadily increasing. Now we can identify a number of more or less competitive systems and new solutions are continuously appearing on the scene, to confirm the maturity of technology and the application.

- Auto-Find from Auto-Vu: (Canada): a system solution originally developed for parking control with visible light solutions; more recently it has been modified with infrared light source to work in all environmental conditions.
- Mobile system from Appian Technology (UK): the early solution was based on passive vision using standard video sensors inside the vehicle; more recently a new configuration based on infrared LED sensor has been introduced in the market.
- Mobile systems from PIPS Technology (UK): a high-quality IR integrated sensory system has been developed for mobile applications to work both at high speed on the highway and at low-speed in downtown.

4 Discussion and Conclusion

Unfortunately, not always the most exciting research subjects bring to business success. A favourable combination of different factors is required, including a significant component of technology innovation, the identification of all relevant

added values to the market, as well as a solution to real concrete problems. R&D technology investment may become extremely profitable when are oriented to problem solving. Robustness and reliability and high quality of the results are also fundamental issues (product engineering and system configuration management) as well as the subjective quality (as perceived by the user). The solution must be effectively integrated within existing operational procedure and particular care must be devoted to all services of maintenance and user assistance. The technology development team must exhibit a high flexibility to adapt the proposed solution to user requirements along the development of the product, with the involvement of the user in the development phase. The company qualification and certification is extremely important. Credibility and reputation are difficult to achieve but they are very easy to lose. In this context IPR activity (patents & publications) as well as licensing strategy must be well defined and supported.

Partnership relationships are extremely critical in the development of a new business based on R&D innovation. It may be technical (to complement internal competence and "core" technology) as well as commercial (to support introduction especially in the international market).

The referred example of Auto-Detector is a successful combination of both technical and commercial components where the maturity of the Imaging Technology (Plate Recognition in real-time) and the nice idea to let it free to move inside the traffic flow to automatically search for the plates in the list, was properly fitting the requirements of security institutions in the new emerging field of Homeland Security.

Any new project is a different story by itself but I do hope t this experience to be of some help for a better understanding of the complex issue of R&D Technology development. Moreover it may be a stimulus for the development of new and more ambitious projects based on Computer Vision.

References

1. W.B. Rouse, "R&D/Technology Management: a framework for Putting Technology to Work", IEEE Trans. On Systems, Man and Cybernetics, part C: applications and Reviews, vol 28, n.4 Nov. 1998, pp.501-515.
2. T. J. Allen "Managing the flow of Technology", Cambridge, MA; MIT Press, 1977.
3. The European Information Society Technologies Prize, EEC IST-prize, www.ist-prize.org.
4. T.K. Sung, B.S Kang, S K Lee, "Study on Characteristics of Technology Transfer in Venture Business", Proceedings of the 34th Hawai Int. Conf. On System Sciences, 2001.
5. G.Garibotto, "Auto-Detector: Mobile Automatic Number Plate Recognition", Handbook of Pattern Recognition and Computer Vision, ed. C.H.Chen & P.S.P. Wang, chap 5.6, pp. 601-618
6. RELES, Remington-Elsag Law Enforcement System, http://www.remington-elsag.com/Home.htm,

Biometric Recognition: How Do I Know Who You Are?

Anil K. Jain

Department of Computer Science and Engineering,
Michigan State University, MI 48824, U.S.A.
jain@cse.msu.edu

Abstract. Reliable person recognition is an integral component of identity management systems. Biometrics offers a natural and reliable solution to the problem of identity determination by recognizing individuals based on their physiological and/or behavioral characteristics that are inherent to the person. Although biometric systems have been successfully deployed in a number of civilian applications, current biometric systems are not perfect. In this paper, we describe the various obstacles that prevent biometric systems from achieving foolproof automatic person recognition. We also show that using multiple biometric modalities can alleviate some of the problems faced by unimodal biometric systems. Finally, we present the vulnerabilities of biometric systems and discuss solutions to protect biometric systems from some common attacks.

1 Biometric Recognition

A wide variety of systems require reliable person recognition schemes to either confirm or determine the identity of an individual requesting their services. The purpose of such schemes is to ensure that the rendered services are accessed only by a legitimate user, and not anyone else. Examples of such applications include secure access to buildings, computer systems, laptops, cellular phones and ATMs. In the absence of robust person recognition schemes, these systems are vulnerable to the wiles of an impostor. Biometric recognition, or simply biometrics, refers to the automatic recognition of individuals based on their physiological and/or behavioral characteristics. By using biometrics it is possible to confirm or establish an individual's identity based on "who she is", rather than by "what she possesses" (e.g., an ID card) or "what she remembers" (e.g., a password). Although biometrics emerged from its extensive use in law enforcement to identify criminals, i.e., forensics, it is being increasingly used today to carry out person recognition in a large number of civilian applications (e.g., national ID card, e-passport and smart cards) [1], [2]. Most of the emerging applications can be attributed to increased security threats as well as fraud associated with various financial transactions (e.g., credit cards). Another emerging application of biometrics is in the domain of digital rights management. The utilization of digital techniques in the creation, editing and distribution of multimedia data offers a

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 19–26, 2005.

number of opportunities to a pirate user, such as high fidelity copying. Furthermore, Internet is providing additional channels for a pirate to quickly and easily distribute the copyrighted digital content without the fear of being tracked. As a result, the protection of multimedia content (image, video, audio, etc.) is now receiving a substantial amount of attention. Multimedia content protection that is based on biometric data of the users is being investigated [3]. Password-only encryption schemes are vulnerable to illegal key exchange problems. By using biometric data along with hardware identifiers such as keys, it is possible to alleviate fraudulent usage of protected content [4].

What biological measurements qualify to be a biometric? Any human physiological and/or behavioral characteristic can be used as a biometric characteristic as long as it satisfies the following requirements:

- Universality: each person should have the characteristic;
- Distinctiveness: any two persons should be sufficiently different in terms of the characteristic;
- Permanence: the characteristic should be sufficiently invariant (with respect to the matching criterion) over a period of time;
- Collectability: the characteristic can be measured quantitatively.

However, in a practical biometric system (i.e., a system that employs biometrics for person recognition), there are a number of other issues that should be considered, including:

- Performance, which refers to the achievable recognition accuracy and speed, the resources required to achieve the desired performance, as well as the operational and environmental factors that affect the performance;
- Acceptability, which indicates the extent to which people are willing to accept the use of a particular biometric identifier in their daily lives;
- Circumvention, which reflects how easily the system can be fooled using fraudulent methods.

A practical biometric system should meet the specified recognition accuracy, speed, and resource requirements, be harmless to the users, be accepted by the intended population, be easy to use and be sufficiently robust to various fraudulent methods and attacks on the system. Among the various biometric measurements in use, fingerprint-based systems [5] and face recognition systems [6] are the most popular.

A biometric system is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database. Depending on the application context, a biometric system may operate either in a verification mode or an identification mode. A biometric system is designed using the following four main modules: (i) sensor module, (ii) feature extraction module, (iii) matcher module, and (iv) system database module.

The response of a biometric system is a matching score that quantifies the similarity between the input and the database template representation. Higher

score indicates that the system is more certain that the two biometric measurements come from the same person. The system decision is regulated by the threshold: pairs of biometric samples generating scores higher than or equal to the threshold are inferred as mate pairs (i.e., belonging to the same person); pairs of biometric samples generating scores lower than the threshold are inferred as non-mate pairs (i.e., belonging to different persons). A biometric verification system makes two types of errors: (i) mistaking biometric measurements from two different persons to be from the same person (called *false match*), and (ii) mistaking two biometric measurements from the same person to be from two different persons (called *false non-match*). These two types of errors are often termed as *false accept* and *false reject*, respectively.

Deployment of biometric systems in various civilian applications does not imply that biometric recognition is a fully solved problem. Table 1 presents the state-of-the-art error rates of three popular biometric traits. It is clear that there is a plenty of scope for improvement in the performance of biometric systems. We not only need to address issues related to reducing error rates, but we also need to look at ways to enhance the usability of biometric systems and address the *return on investment* issue.

Table 1. State-of-the-art error rates associated with fingerprint, face and voice biometric systems. Note that the accuracy estimates of biometric systems are dependent on a number of test conditions.

	Test	**Test Parameter**	**False Reject Rate**	**False Accept Rate**
Fingerprint	FVC 2004 [7]	Exaggerated skin distortion, rotation	2%	2%
	FpVTE 2003 [8]	U.S. government operational data	0.1%	1%
Face	FRVT 2002 [9]	Varied lighting, outdoor/indoor	10%	1%
Voice	NIST 2004 [10]	Text independent, multi-lingual	5-10%	2-5%

2 Multimodal Biometrics

Biometric systems that perform person recognition based on a single source of biometric information are often affected by the following problems [11]:

– Noisy sensor data : Noise can be present in the acquired biometric data mainly due to defective or improperly maintained sensors. For example, accumulation of dirt or the residual remains on a fingerprint sensor can result in a noisy fingerprint image. The recognition accuracy of a biometric system is highly sensitive to the quality of the biometric input and noisy data can result in a significant reduction in the accuracy of the biometric system [12].

- Non-universality: Not all biometric traits are truly universal. The National Institute of Standards and Technology (NIST) has reported that it is not possible to obtain a good quality fingerprint from approximately two percent of the population (people with hand-related disabilities, manual workers with many cuts and bruises on their fingertips, and people with oily or dry fingers) [13]. Hence, such people cannot be enrolled in a fingerprint verification system. Similarly, persons having long eye-lashes and those suffering from eye abnormalities or diseases cannot provide good quality iris images for automatic recognition [14]. Non-universality leads to Failure to Enroll (FTE) and/or Failure to Capture (FTC) errors in a biometric system.
- Lack of individuality: Features extracted from biometric characteristics of different individuals can be quite similar. For example, appearance-based facial features that are commonly used in most of the current face recognition systems are found to have limited discrimination capability [15]. A small proportion of the population can have nearly identical facial appearance due to genetic factors (e.g., father and son, identical twins, etc.). This lack of uniqueness increases the False Match Rate (FMR) of a biometric system.
- Lack of invariant representation: The biometric data acquired from a user during verification will not be identical to the data used for generating the user's template during enrollment. This is known as "intra-class variation". The variations may be due to improper interaction of the user with the sensor (e.g., changes due to rotation, translation and applied pressure when the user places his finger on a fingerprint sensor, changes in pose and expression when the user stands in front of a camera, etc.), use of different sensors during enrollment and verification, changes in the ambient environmental conditions (e.g., illumination changes in a face recognition system) and inherent changes in the biometric trait (e.g., appearance of wrinkles due to aging or presence of facial hair in face images, presence of scars in a fingerprint, etc.). Ideally, the features extracted from the biometric data must be relatively invariant to these changes. However, in most practical biometric systems the features are not invariant and therefore complex matching algorithms are required to take these variations into account. Large intra-class variations usually increase the False Non-Match Rate (FNMR) of a biometric system.
- Susceptibility to circumvention: Although it is difficult to steal someone's biometric traits, it is possible for an impostor to circumvent a biometric system using spoofed traits. Studies [16] have shown that it is possible to construct gummy fingers using lifted fingerprint impressions and utilize them to circumvent a biometric system. Behavioral traits like signature and voice are more susceptible to such attacks than physiological traits. Other kinds of attacks can also be launched to circumvent a biometric system [17].

Some of the problems that affect unimodal biometric systems can be alleviated by using multimodal biometric systems [18]. Systems that consolidate cues obtained from two or more biometric sources for the purpose of person recognition are called multimodal biometric systems. Multimodal biometric systems have several advantages over unimodal systems. Combining the evidence

obtained from different modalities using an effective fusion scheme can significantly improve the overall accuracy of the biometric system. A multimodal biometric system can reduce the FTE/FTC rates and provide more resistance against spoofing because it is difficult to simultaneously spoof multiple biometric sources. By asking the user to present a random subset of biometric traits (e.g., right index finger followed by right middle finger), the system ensures that a "live" user is indeed present at the point of data acquisition. Thus, a challenge-response type of authentication can be facilitated by using multimodal biometric systems. However, multimodal biometric systems also have some disadvantages. They are more expensive and require more resources for computation and storage than unimodal biometric systems. Multimodal systems generally require more time for enrollment and verification causing some inconvenience to the user. Finally, the system accuracy can actually degrade compared to the unimodal system if a proper technique is not followed for combining the evidence provided by the different modalities. However, the advantages of multimodal systems far outweigh the limitations and hence, such systems are being increasingly deployed in security-critical applications.

The design of a multimodal biometric system is strongly dependent on the application scenario. A number of multimodal biometric systems have been proposed in literature that differ from one another in terms of their architecture, the number and choice of biometric modalities, the level at which the evidence is accumulated, and the methods used for the information fusion. Fusion at the matching score level is generally preferred due to the presence of sufficient information content and the ease in accessing and combining matching scores. A principled approach to score level fusion is the computation of likelihood ratios based on the estimates of genuine and impostor score distributions [19]. Information obtained from soft biometric identifiers like gender, ethnicity and height can also be integrated with the primary biometric information like face and fingerprint, to improve the recognition accuracy of the biometric system [20].

3 Biometric System Vulnerabilities

Biometric systems are vulnerable to several kinds of attacks that can compromise the security afforded by the biometric component and causing the failure of the system that it is intended to protect. Figure 1 summarizes the ways in which a biometric system can be attacked. The failure of a biometric system can be classified into two types. Large inter-class and small inter-class variability may result in the matcher erroneously accepting an impostor. Studies on the individuality of the biometric trait attempt to calculate the theoretical probability of this type of biometric system failure. For example, individuality of the minutiae information in a fingerprint was studied in [21].

The second kind of biometric system failure occurs when an impostor is deliberately attempting to masquerade the system. Ratha et al. [17] identified different levels of attacks that can be launched against a biometric system. These attacks are intended to either circumvent the security afforded by the system or

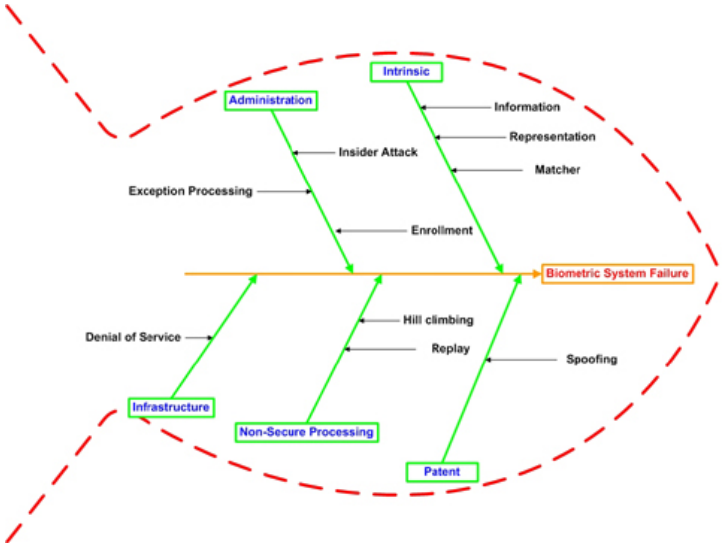


Fig. 1. Fishbone (cause & effect) illustration of biometric failures. The security afforded by a biometric system can be undermined by: (a) Administration: The system administrator can abuse and compromise the system. (b) Intrinsic: The inherent limitations of the representation/matching schemes may result in erroneously accepting an intruder. (c) Infrastructure: Denial of service attacks can disable system functionality. (d) Non-secure processing: An impostor can hack system processes to gain access into the system. (e) Patent: Since biometric identifiers are not secrets, an impostor could create physical or digital artifacts to fool the system.

to deter the normal functioning of the system: (i) A fake biometric trait such as an artificial finger may be presented at the sensor. (ii) Illegally intercepted data may be resubmitted to the system. (iii) The feature extractor may be replaced by a Trojan horse program that produces pre-determined feature sets. (iv) Legitimate feature sets may be replaced with synthetic feature sets. (v) The matcher may be replaced by a Trojan horse program that always outputs high scores thereby defying the system security. (vi) The templates stored in the database may be modified or removed. Alternately, new templates may be introduced in the database. (vii) The data in the communication channel between various modules of the system may be altered. (viii) The final decision output by the biometric system may be overridden.

Among these attacks, the presentation of fake biometric traits at the sensor and the protection of biometric templates have been widely studied in the literature and a number of solutions have been proposed to guard against such attacks. A challenge-response type of authentication can prevent the problem of fake biometric submission to a great extent. Other methods such as detection of liveness during the presentation of the biometric trait have also been suggested.

For the protection of biometric templates, Jain and Uludag [22] suggested the use of steganography principles that hide biometric data (e.g., eigen-coefficients of a face image) in host images (e.g., fingerprints). Ratha et al. [23] proposed the use of distortion functions to generate biometric data that can be canceled if necessary. Thus, careful design and planning is necessary to ensure the integrity of the biometric system and thwart the impostor attempts to circumvent the security of the system.

4 Summary

Reliable person recognition is critical to many government and business processes. The conventional knowledge-based and token-based methods do not really provide positive person recognition because they rely on surrogate representations of the person's identity (e.g., exclusive knowledge or possession). It is, thus, obvious that any system assuring reliable person recognition must necessarily involve a biometric component. This is not, however, to state that biometrics alone can deliver error-free person recognition. In fact, a sound system design will often entail incorporation of many biometric and non-biometric components (building blocks) to provide reliable person recognition. As biometric technology matures, there will be an increasing interaction among the market, technology, and the applications. This interaction will be influenced by the added value of the technology, user acceptance, and the credibility of the service provider. It is too early to predict where and how biometric technology would evolve and get embedded in which applications. But it is certain that biometric-based recognition will have a profound influence on the way we conduct our daily business.

References

1. Jain, A.K., Ross, A., Prabhakar, S.: An Introduction to Biometric Recognition. IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics **14** (2004) 4–20
2. Wayman, J.L., Jain, A.K., Maltoni, D., Maio, D., eds.: Biometric Systems, Technology, Design and Performance Evaluation. Springer (2005)
3. Uludag, U., Jain, A.K.: Multimedia Content Protection via Biometrics-based Encryption. In: Proceedings of IEEE International Conference on Multimedia and Expo, vol. III, Baltimore, USA (July 2003) 237–240
4. Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.K.: Biometric Cryptosystems: Issues and Challenges. Proceedings of IEEE, Special Issue on Multimedia Security for Digital Rights Management **92** (2004) 948–960
5. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer (2003)
6. Li, S., Jain, A.K., eds.: Handbook of Face Recognition. Springer (2005)
7. Maio, D., Maltoni, D., Cappelli, R., Wayman, J.L., Jain, A.K.: FVC2004: Third Fingerprint Verification Competition. In: Proceedings of International Conference on Biometric Authentication, Hong Kong, China (2004) 1–7

8. Wilson, C., Hicklin, A.R., Korves, H., Ulery, B., Zoepfl, M., Bone, M., Grother, P., Micheals, R.J., Otto, S., Watson, C.: Fingerprint Vendor Technology Evaluation 2003: Summary of Results and Analysis Report. NIST Internal Report 7123; available at `http://fpvte.nist.gov/report/ir_7123_summary.pdf` (2004)
9. Philips, P.J., Grother, P., Micheals, R.J., Blackburn, D.M., Tabassi, E., Bone, J.M.: FRVT2002: Overview and Summary. Available at `http://www.frvt.org/FRVT2002/documents.htm` (2002)
10. Reynolds, D.A., Campbell, W., Gleason, T., Quillen, C., Sturim, D., Torres-Carrasquillo, P., Adami, A.: The 2004 MIT Lincoln Laboratory Speaker Recognition System. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. Volume 1., Philadelphia, PA (2005) 177–180
11. Jain, A.K., Ross, A.: Multibiometric Systems. Communications of the ACM, Special Issue on Multimodal Interfaces **47** (2004) 34–40
12. Chen, Y., Dass, S.C., Jain, A.K.: Fingerprint Quality Indices for Predicting Authentication Performance. In: Proceedings of Fifth International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA) (To appear), New York, U.S.A. (2005)
13. NIST report to the United States Congress: Summary of NIST Standards for Biometric Accuracy, Tamper Resistance, and Interoperability. Available at `ftp://sequoyah.nist.gov/pub/nist_internal_reports/NISTAPP_Nov02.pdf` (2002)
14. News, B.: Long lashes thwart ID scan trial. Available at `http://news.bbc.co.uk/2/hi/uk_news/politics/3693375.stm` (2004)
15. Golfarelli, M., Maio, D., Maltoni, D.: On the Error-Reject Tradeoff in Biometric Verification Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 786–796
16. Matsumoto, T., Matsumoto, H., Yamada, K., Hoshino, S.: Impact of Artificial "Gummy" Fingers on Fingerprint Systems. In: Optical Security and Counterfeit Deterrence Techniques IV, Proceedings of SPIE. Volume 4677. (2002) 275–289
17. Ratha, N.K., Connell, J.H., Bolle, R.M.: An Analysis of Minutiae Matching Strength. In: Proceedings of Third International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA), Sweden (2001) 223–228
18. Hong, L., Jain, A.K., Pankanti, S.: Can Multibiometrics Improve Performance? In: Proceedings of IEEE Workshop on Automatic Identification Advanced Technologies, New Jersey, U.S.A. (1999) 59–64
19. Dass, S.C., Nandakumar, K., Jain, A.K.: A Principled Approach to Score Level Fusion in Multimodal Biometric Systems. In: Proceedings of Fifth International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA) (To appear), New York, U.S.A. (2005)
20. Jain, A.K., Nandakumar, K., Lu, X., Park, U.: Integrating Faces, Fingerprints and Soft Biometric Traits for User Recognition. In: Proceedings of Biometric Authentication Workshop, LNCS 3087, Prague, Czech Republic (2004) 259–269
21. Pankanti, S., Prabhakar, S., Jain, A.K.: On the Individuality of Fingerprints. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 1010–1025
22. Jain, A.K., Uludag, U.: Hiding Biometric Data. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 1493–1498
23. Ratha, N., Connell, J., Bolle, R.: Enhancing Security and Privacy in Biometrics-based Authentication Systems. IBM Systems Journal **40** (2001) 614–634

Unsupervised Symbol Grounding and Cognitive Bootstrapping in Cognitive Vision

R. Bowden, L. Ellis, J. Kittler, M. Shevchenko, and D. Windridge

Centre for Vision, Speech and Signal Processing,
The University of Surrey, Guildford, Surrey, GU2 7XH, UK

Abstract. In conventional computer vision systems symbol grounding is invariably established via supervised learning. We investigate unsupervised symbol grounding mechanisms that rely on perception action coupling[1]. The mechanisms involve unsupervised clustering of observed actions and percepts. Their association gives rise to behaviours that emulate human action. The capability of the system is demonstrated on the problem of mimicking shape puzzle solving. It is argued that the same mechanisms support unsupervised cognitive bootstrapping in cognitive vision.

1 Introduction

Computer Vision as a branch of Artificial Intelligence (AI) has, over recent years diverged from its roots. Forming a science within itself, it relies heavily upon statistical techniques. The recent interest in Cognitive Vision is, to some extent, an attempt to reunite computer vision with its AI roots.

Some argue that all computer vision is cognitive and the mere embodiment of a statistical technique, for example in robot vision, produces a cognitive system. Others actively try and combine modern day computer vision with more traditional AI approaches. However, the fusion of statistical and symbolic information raises a whole host of questions: What is the meaning of a symbol? How is it grounded? How is it constructed? Should symbols emerge or are the provided to the system a priori? Should symbols change/evolve? What syntax governs them and how can we learn this syntax irrespective of application? How can the system bootstrap itself to enhance its perceptual and reasoning capabilities?

The fundamental flaw in most cognitive vision development is its tailoring to specific applications. Transferable learning is a key aspect of cognitive approaches but a successful approach should provide generic learning, where, the same framework can be applied to a whole class of problems. This paper addresses some of the issues and presents early work towards generic symbolic processing and bootstrapping mechanisms for cognitive systems.

Following a discussion on symbol grounding in visual agents in Section 2, we present a novel unsupervised symbol grounding mechanism based on percept clustering and perception-action coupling in Section 3. Section 4 addresses the issues of cognitive bootstrapping. Finally conclusions are drawn in Section 5.

[1] This work was supported by EU projects COSPAL and VAMPIRE

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 27–36, 2005.

2 Symbol Grounding

A central practical and philosophical problem for the construction of autonomous visual agents is that of *symbol grounding* [8]. An autonomous visual agent is, by definition, one capable of adapting to its in environment in behavioural and representational terms that go beyond those implied by its initial set of 'bootstrap' symbolic assumptions. Doing so necessitates the use of mechanisms of generalisation, inference and decision making in order to modify the initial perceptual symbol set in the light of novel forms of sensory data.

Any visual representation capable of abstract generalisation is implicitly governed by the laws of predicate logic. As such, the generalised entities must observe strictly formalised laws of interrelationship, and consequently, in abstracting the visual symbol set away from the original set of innate percept-behavioural pairings, are apt to become detached from any intrinsic meaning in relation to the agent's environment. A related difficulty, known as the *frame problem* [17], also arises in such generalised formal domains; it is by no mean clear which particular set of logical consequences (given the infinite number of possibilities) that the generalised reasoning system should concern itself with.

There is therefore a problem of symbol relevance and 'grounding' unless additional mechanisms can be put in place to form a bridge between the formal requirements of logical inference (applied to visual symbols), and the relevance of this symbol set to the agent within the context of both its goals and the intrinsic nature of the environment. In terms of the philosophy of cognition, this necessitates a move from a Quinean [9] to a Wittgensteinian [10] frame of reference, in which symbol *meaning* is intrinsically contextual, and environment-dependent.

For artificial cognitive agents embodied within the real world (that is to say, *robots*), the form that this symbol grounding framework must take is, by an increasing consensus (eg [11], [12], [13], [21]), one of hierarchical stages of abstraction that proceed from the 'bottom-up'. At the lowest level is thus the immediate relationship between percept and action; a change in what is perceived is primarily brought about by actions in the agent's motor-space. This hence limits visual learning to what is immediately relevant to the agent, and significantly reduces the quantity of data from which the agent must construct its symbol domain by virtue of the many-to-one mapping that exists between the pre-symbolic visual space and the intrinsic motor space [14]. For example, a hypothetical mobile robot engaged in simultaneous location and mapping (SLAM) (eg [16]) might build up a stationary stochastic model of any environmental changes that occur when not engaged in any direct motor activity, but switches to a Markovian transitional model when engaged in motor activity, thereby forming a sequence of 'key-frame' transitions driven by its motor impulses.

The first level of abstraction in the hierarchy thus represents a generalisation of the immediate, pre-symbolic percept-action relation into the *symbol domain*. There are many approaches to achieving this primary generalisation, for instance: unsupervised clustering [14], invariant subspace factoring [18], constructive solid geometry schematics [19]. Progressive levels of abstraction can be added by similar means, or they might instead involve higher levels of inferential machinery,

for instance first order logical induction for rule inference, if explicitly ascending the Chomsky hierarchy [15].

At some level of abstraction, critically, is the concept of *objects*, characterised by their persistence with respect to the agent's actions. Representations above this level are then characterised by their object-centric, rather than agent-centric descriptions (so we move from a percept-action space into a domain where descriptions with formal equivalents to English terms such as '*on*', '*under*', etc, can form part of the environment description). What results is a set of high-level, abstracted symbol generalisations that are nevertheless grounded in the percept-action space by virtue of the intermediate hierarchical levels. We might thus, for instance, envisage a tennis-playing robot that has the segmentation of the ball from the background at its lowest representative level, leading into a series of ascending representations that cumulate in the formal logical rules of the game of tennis at the most abstract level of representation. Furthermore, such a hierarchical structure has the advantage that higher-level action imperatives (such as, in our example, 'serving the ball') may act to reinforce learning at the lower-levels (by providing additional tennis-ball segmentation statistics). The hierarchical percept-action structure is hence robust and adaptive by nature.

A further possibility, once higher-levels of the perception-action hierarchy are sufficiently well established, is (by way of contrast to the previous passive example) to use these to *actively* drive lower-level learning. Hence, an inferred *partial* environment representation in an autonomous mobile robot might be used to initiate exploration into unmapped regions of the environment, or to improve upon weakly mapped environmental domains. Alternatively, percept clustering can itself be driven by higher-level concepts inferred from *those same* clusters, such as in Magee *et al.*'s [14] visual first-order logic induction system, in which clustered entities with identical inferred logical relations are deemed to be the *same* (which is to say they are meta-clustered by the higher-level inferential system).

Like the first example, this latter approach thus has the capacity to completely solve one of the critical difficulties of unsupervised cognitive learning that we have alluded to; the issue of *framing*. By deciding at what level to cluster entities in the sensory domain on the basis of entities formed from those same clusters, the potential for redundant higher-level inferences from the sensory domain can be entirely eliminated. The symbol structure thus becomes *entirely* agent-relative, irrespective of the initial set of symbolic assumptions with which the system was 'bootstrapped' into cognitive activity. We hence term this active hierarchical-feedback approach to autonomous cognition **'cognitive bootstrapping'** by virtue of the capacity of the cognitive systems so described to make their symbolic representations fully *self-foundational*.

It is apparent that classical AI approaches to cognitive vision were unsuccessful in that they attempted to build a high-level environmental description *directly from* the percept space before going on to consider agent actions within this model, rather than allowing this representation to evolve at a higher hierarchical level [20]. Representative priorities were thus specified in advance by the

system-builder and not by the agent, meaning that *autonomous* agency had to build its goals and higher-level representations in terms of the *a priori* representation, with all the redundancy that this implied. Furthermore, novel modes of representation were frequently ruled out in advance by the pre-specification of scene-description.

In the following sections we shall discuss a few mechanisms that accomplish symbol grounding without conventional learning in a supervised mode. The symbol grounding is achieved by associating percepts with actions. This association gives rise to interesting perception - action behaviours. As an example, in the next section, we demonstrate that the system can learn to play a game such as puzzle. More interestingly, we show that unsupervised clustering and/or quantisation of percepts and observed actions lead to the discovery of new concepts and functionalities, characteristic of bootstrap learning and emergence of intelligence.

3 Modelling Perception Action Coupling

In this section we present a framework for autonomous behaviour in vision based artificial cognitive systems by imitation through coupled percept-action (stimulus and response) exemplars.

The assumption is made that if a cognitive system has, stored in its memory, a representation of all the possible symbolic perceptual stimuli (percepts) that it shall ever encounter each coupled with a 'symbolic' action model (response), then it should be capable of responding as required to any given visual stimulus. This assumption leads us to consider how a practical estimation to such a system could be achieved.

The system must be capable of searching its entire percept store (visual memory) in order to find a match to the current percept (visual stimulus), and then perform the associated action.

By hierarchically grouping the percepts into progressively more general representations(expressions), and given that the stored percepts adequately cover the percept space, we can structure the stored percepts in such a way as to allow fast searching of the percept space. Also in generalising the percept representations further at each level of the hierarchy, the system is capable of compensating for the incomplete coverage of the percept space by performing more general action models given more general percept representations.

This system operates within a simulated shape sorter puzzle environment. The training phase is initiated by the supervisor solving the shape sorter puzzle a number of times. Each time the supervisor takes some action, the system records both the action performed and the associated visual percept. During the systems on-line state it extracts the current scene information, builds a symbolic representation in order to compare to the stored percepts and then performs the action associated with the best matching percept.

In order to cluster percepts at each level of the hierarchy, the representation must allow us to measure similarity between two percepts. In this system a scene

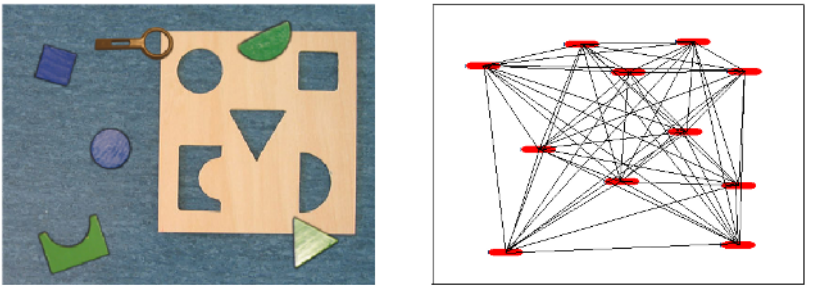


Fig. 1. Percepts are represented as Attributed Relational Graphs

is represented symbolically as an Attributed Relational Graph. Graph vertices represent the objects in the scene. Graph edges represent the relational structure of the scene, see figure-1. *Type* attributes are attached to each vertex and dictate the type of object. Graph edge attributes are 3D *relative_position/orientation* vectors that represent both the horizontal and vertical displacement and the relative orientation between the two objects connected by the edge.

Formally we define Attributed Relational Graphs (ARGs) as a 4-tuple, g = (V,E,u,v) where V and E (E $\subseteq$ V x V) are the set of nodes (graph vertexes) and edges (links between nodes) respectively. u : V $\longrightarrow$ Av is a function assigning attributes to nodes, and v : E $\longrightarrow$ Ae is a function assigning attributes to edges. Av and Ae are the sets of node and edge attributes respectively.

In order to group the percepts we need some way to measure/compute the similarity or distance between two Attributed Relational Graphs. We have adopted the VF graph matching algorithm [7] developed at the Artificial Vision Group (University of Naples). The median of each group/cluster is computed and is used to represent the cluster members at the current level. The median graph is computed by finding the graph that has the minimum sum of distances to all other cluster members.

In order for a cognitive system to actively respond to visual stimulus, a mapping between percepts and actions is required [1]. Recent neurophysiological research has shown strong evidence supporting the existence of a mechanism, in both primate and human brains, that obtains this percept-action coupling, known in lit. as "direct-matching hypothesis". The mirror-neuron system essentially provides the system (human/primate brain) with the capability "to recognise actions performed by others by mapping the observed action on his/her own motor representation of the observed action" [2].The system presented here has been endowed with these same innate capabilities. Therefore our work differs from that of [3] where an attempt is made to analyse from visual data, the force dynamics of a sequence and hence deduce the action performed. Instead, by allowing the system to directly record symbolic representations of the actions performed during the training phase, an exact percept-action coupling can be achieved. As an alternative approach, [4] has shown that it is possible for an agent to learn to mimic a human supervisor by first observing simple tasks and then, through experimentation, learning to perform the actions that make up the tasks.

h

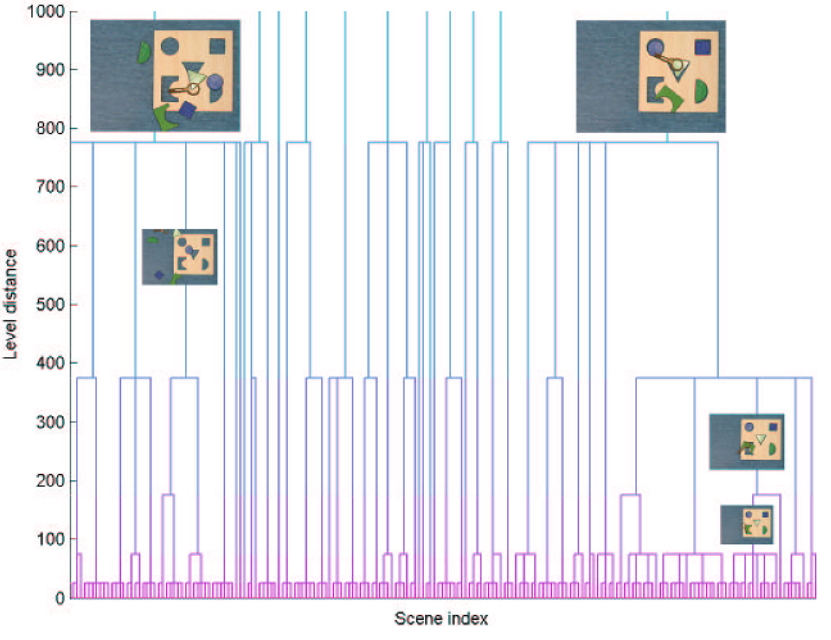


Fig. 2. Percepts are clustered into a hierarchy

The lowest level actions that the system is capable of performing are 'pick up object', 'put down object' and 'move gripper'. Within the context of the problem the system needs only to perform these actions in a fixed format: A:= move - pick up - move - put down. Since some of the intermediary percept-action pairs are no longer needed (e.g. those percepts coupled to the 'pick up' or 'put down' action), *key scenes* corresponding to the beginning of a sequence, must be extracted and coupled with the action model. It is these key scenes, represented as ARGs, that form the data set of the systems visual memory. It is worth making clear here that we have temporally segmented the continuous perceptual input in accordance with the beginning of our fixed format action sequences and discarded all the perceptual data not extracted as key scenes. This approach has the advantage of reducing the amount of data needed to store an entire puzzle sequence.

Another motivation for fixing the action format is that a fixed length vector can now be used to represent an action. To model the actions that are coupled to the key scenes, only a five-element vector is required. The first and second elements in the action vector represent the change, brought about by the first 'move' action, in the horizontal and vertical positions respectively. As the 'pick up' and 'put down' parts of the action need no parameterisation and are implicit to the action, they are not represented in the action vector. The last three elements of the action vector are therefore left to parameterise the final 'move' operation. This is the same as the first 'move' but with a third 'rotation' dimension.

Now that an action can be modelled as a vector, the actions coupled to a cluster of percepts can be represented as a matrix. Note that the action clusters are a result of perceptual grouping.

Many of the activities that cognitive systems must perform require perceptually guided action e.g. walking or catching. It is also true to say that much of what is perceived by a cognitive system is the result of actions performed by that system e.g. moving to a new space in the environment or picking up an object to examine it. The perception-action cycle simply describes a model of behaviour whereby perception influences action and action influences perception.

The system extracts a symbolic representation, percept, from the current sensory input. It then finds the closest matching percepts in its visual-proprioceptive memory by searching the percept hierarchy. The resulting matches are each associated with an action vector and, depending on the level in the hierarchy at which the match is made, a generality parameter. The action is performed and so the scene changes and once again the system extracts a new percept and so the cycle continues until a solution is reached. The operator must *teach* the system by solving the puzzle a number of times. This provides the system with the expected behaviour that it will attempt to imitate during game play.

4 Cognitive Bootstrapping

The framework described in the previous section endows the system with the ability to imitate action behaviours. [5] have recently presented a framework for learning of object, event and protocol models from audio visual data. Their framework employs both statistical learning methods, for object learning, and symbolic learning for sequences of events representing implicit temporal protocols. As [5] point out this learning of temporal protocols is analogous to grammar learning, in this respect the system presented here shares some goals with that presented by [5]. Further, both systems attempt to achieve this grammar learning through generalising a symbolic data set. There is however very little similarity between the approach taken by [5]; Inductive Logic Programming, and that which we have taken; clustered percept-action exemplars. Where [5] have developed, using Progol, an inference engine to extract temporal protocols, we have employed an *approximation to imitation* approach to learning puzzle grammars/temporal protocols.

It should be noted that we have given the system no indication as to what the goal of the activity is and what represents a successful conclusion of the game. In fact without any other functionalities the system would be unable to change its behaviour and reason about unusual states. To correct this, the system needs a mechanism that can discover the goal of an observed activity. Once the goal is defined, the behaviour can be optimised or even adapted by direct optimisation of the objective function that encapsulates the activity goal. The goal can be discovered by clustering observed actions or by clustering the perceptual stimuli corresponding to the activity end state. The *solved puzzle* state is recorded by the system each time the user finishes the puzzle. Such perceptual data analysis

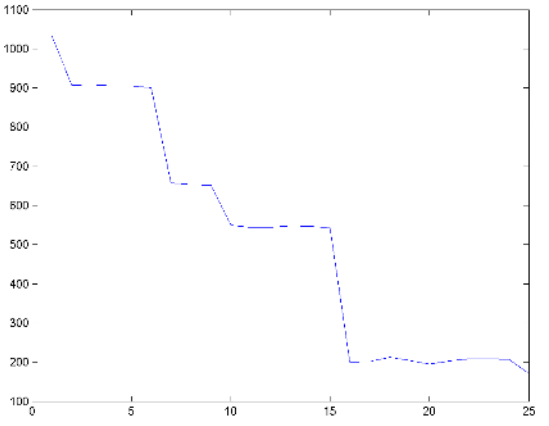


Fig. 3. The cost function plot for a successful sequence of actions leading to a solution

would show the end state to be unique and reaching it must be the goal of the behaviour. The definition of the goal then allows a cost function to be set up in terms of distance to end state.

In order to allow the system to improve its performance at a given task over time, during its own interactions with the world, it stores the resulting percept-action pairs. The system is capable of supervising itself over time by rewarding near optimal action sequences. Optimality is defined in terms of the distance between the current percept and a percept relating to a *solved puzzle* state.

In addition to self optimisation, the existence of the goal leads to a new type of behaviour where at each step of the game an action minimising the distance to the end state is selected from the list of available actions. The monotonic behaviour of the cost function for a sequence of actions leading to a solution is shown in figure 3. An equally effective solution could be found by clustering actions. This *hard wired* ability to explore the action space in the direction of its modes or in the directions of the modes of the percept space is considered to be innate. It is instrumental in providing bootstrapping capability, as it enables the system to discover new solutions and to reason about the task.

At this point it is pertinent to ask, how the vision system could bootstrap itself when it is first switched on and has no prior perceptual capabilities. The above shape puzzle problem has been solved under the assumption that the system can segment objects by colour and shape and is able to move its arm/gripper intentionally from point to point, as well as pick and place objects. Unless one opts for hard wiring, such functionalities have to be acquired by self learning.

We have studied this problem and shown that perception action cycle, combined with the unsupervised learning and action space exploration mechanisms discussed above are sufficient to build up vision capabilities from the ground level. The only assumption we make is that the system is able to drive its arm, in a random manner and observe its movement. This has been simulated by representing the arm as a single point in a 2D work space and moving it from one

point to another. The random moves were constrained by hard environmental boundaries. Such constraints are realistic, as they emulate the world being dominantly horizontal, subject to the laws of gravity. Thus random motions will often be confined to horizontal and vertical directions imposed by the constraints.

Unsupervised clustering of the observed visual data detect these dominant directions which then activate the exploration module to attempt arm movements in the required directions. As the system, initially, has no link between action and perception, the dominant directions in the percept space provide an action goal the achievement of which can be measured in the percept space. This internal goal of the system and the observed error allow the system to learn to perform the intentional actions effectively in a supervised mode.

Once the system is self trained to move its arm in the two basic directions, the system discovers that driving the arm will result in its displacement and that the distance travelled horizontally or vertically will depend on the strength of the driving force (frequency and duration of the motor neuron signal). Also, the system is now able to move from point A to point B in a city block manner. By self optimising, the system eventually acquires the ability to move intentionally from any point in its work space to any other and to establish a link between perception and action. The acquisition of other low level vision and action capabilities in a bootstrapping mode is currently investigated and will be demonstrated by the system in the future.

5 Conclusions

We addressed the problem of symbol grounding in cognitive vision. In conventional computer vision systems symbol grounding is invariably established via supervised learning. However, such approaches make the system too application specific and provide no capability for self learning, self optimisation, acquisition of novel behaviours and general reasoning. We investigated unsupervised symbol grounding mechanisms that relied on perception action coupling. The mechanisms involved unsupervised clustering of observed actions and percepts. Their association gave rise to behaviours that emulated human action. The capability of the system was demonstrated on the problem of mimicking shape puzzle solving. We showed that the same mechanisms supported unsupervised cognitive bootstrapping in cognitive vision.

References

1. Granlund, G. 2003. Organization of Architectures for Cognitive Vision Systems. In *Proceedings of Workshop on Cognitive Vision.*
2. Buccino, G.; Binkofski, F.; and Riggio L. 2004. The mirror neuron system and action recognition. In *Brain and Language, volume 89, issue 2, 370-376.*
3. J. M. Siskind 2003. Reconstructing force-dynamic models from video sequences. In *Artificial Intelligence archive, Volume 151 , Issue 1-2 91 - 154.*

4. P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning About Objects Through Action: Initial Steps Towards Artificial Cognition. In *2003 IEEE International Conference on Robotics and Automation (ICRA).*
5. Magee D., Needham C.J., Santos P., Cohn A.G. and Hogg D.C. Autonomous learning for a cognitive agent using continuous models and inductive logic programming from audio-visual input. In *Proc. AAAI Workshop on Anchoring Symbols to Sensor Data, 17-24.*
6. Nock R. and Nielsen F. Statistical Region Merging. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26, Issue 11, 1452- 1458*
7. Cordella L., Foggia P., Sansone C., Vento M. An ecient algorithm for the inexact matching of arg graphs using a contextual transformational model. In *Proceedings of the International Conference on Pattern Recognition, volume 3, 180-184*
8. Harnad, S. (1990). The symbol grounding problem. *Physica D, 42,* 335-346.
9. Quine, W. von O., 1960. Word and Object. New York: John Wiley and Sons, Cambridge: MIT.
10. Wittgenstein, L., Anscombe, G. E. M. (translator), 'Philosophical investigations : the German text with a revised English translation by Ludwig Wittgenstein', Oxford : Blackwell, 2001, ISBN 0631231277.
11. Marr, D. (1982). Vision. San Francisco: Freeman.
12. Gärdenfors, P. 1994. How logic emerges from the dynamics of information. In Van Eijck/Visser, Logic and Information Flow, 49-77.
13. Granlund G., Organization of Architectures for Cognitive Vision Systems, 2003, Proceedings of Workshop on Cognitive Vision, Schloss Dagstuhl, Germany.
14. D. Magee, C. J. Needham, P. Santos, A. G. Cohn, and D. C. Hogg (2004), Autonomous learning for a cognitive agent using continuous models and inductive logic programming from audio-visual input, AAAI Workshop on Anchoring Symbols to Sensor Data.
15. Chomsky, N., Three models for the description of language, IRE Transactions on Information Theory, 2 (1956), pages 113-124
16. Thrun S.,2002, Robotic Mapping: A Survey, Exploring Artificial Intelligence in the New Millennium. Morgan Kauffmann
17. McCarthy, J. & Hayes, P.J. (1969), Some Philosophical Problems from the Standpoint of Artificial Intelligence, in Machine Intelligence 4, ed. D.Michie and B.Meltzer, Edinburgh: Edinburgh University Press, pp. 463-502.
18. Granlund G. H. and Moe A., Unrestricted Recognition of 3D Objects for Robotics Using Multilevel Triplet Invariants., AI Magazine, vol .25, num. 2, 2004, p51-67
19. A. Chella, M. Frixione, S. Gaglio: A Cognitive Architecture for Artificial Vision, Artif. Intell. 89, No. 1-2, pp. 73-111, 1997.
20. Brooks, R. A., Intelligence without Representation, Artificial Intelligence, Vol.47, 1991, pp.139-159
21. Johnson-Laird P. N., Mental Models, Harvard University Press, Cambridge, MA 1983.

Interactive, Mobile, Distributed Pattern Recognition

George Nagy

DocLab, Rensselaer Polytechnic Institute, Troy, NY USA 12180
nagy@ecse.rpi.edu

Abstract. As the accuracy of conventional classifiers, based only on a static partitioning of feature space, appears to be approaching a limit, it may be useful to consider alternative approaches. Interactive classification is often more accurate then algorithmic classification, and requires less time than the unaided human. It is more suitable for the recognition of natural patterns in a narrow domain like trees, weeds or faces than for symbolic patterns like letters and phonemes. On the other hand, symbolic patterns lend themselves better to using context and style to recognize entire fields instead of individual patterns. Algorithmic learning and adaptation is facilitated by accurate statistics gleaned from large samples in the case of symbolic patterns, and by skilled human judgment in the case of natural patterns. Recent technological advances like pocket computers, camera phones and wireless networks will have greater influence on mobile, distributed, interactive recognition of natural patterns than on conventional high-volume applications like mail sorting , check reading or forms processing.

1 Introduction

I am grateful for this wonderful opportunity to proselytize for some heretical notions. First, I will suggest classifying pattern recognition applications into types A, B, AB, and O, according to the pattern recognition methodology that suits each best. Type A consists of *symbolic patterns*, the glyphs and sounds used for encoding messages. Type B includes *natural objects* like flowers and faces that are not used primarily for communication. I will try to substantiate the claim that interactive computer vision, where the tasks leading to object recognition are assigned according to the relative competence of human and machine, is particularly appropriate for Type B applications. On the other hand, context and style based classification seems better suited to Type A applications. Learning and adaptation benefit every type.

Section 2 outlines the considerations that led to the proposed taxonomy of recognition problems. Section 3 summarizes our recent results on interactive classification of flowers and faces. Section 4 and 5 present the corollaries of interactive classification: mobile and networked recognition. In Section 6 we recapture the notion of style, show that it can lead to more accurate classification of multi-source patterns when the test samples are partitioned by source, and contrast it to the better established methods based on language context. In the last section I list some areas where rapid progress may be possible. This is not a survey: however, the cited publications contain extensive references to invaluable prior work by others.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 37 – 49, 2005.

2 Symbolic and Natural Patterns

Prototypical examples of Type A applications are character, hand print, and speech recognition (OCR, ICR and ASR). The first operational OCR system was installed at Readers' Digest in 1955. Eleven years later, at the 1966 IEEE Pattern Recognition Workshop in Puerto Rico, postal address readers, form processing, and spoken word recognition were among the most popular topics, and they remain so today. The early OCR systems were so expensive that they could not be justified unless they displaced 10-20 keypunch operators. Target rates were 1000-2000 characters per second. High throughput was necessary because our garrulous species spawns endless streams of print that we yearn to preserve for posterity. Type A applications share the following characteristics:

- they deal with symbolic patterns that represent natural or formal languages;
- any reader or speaker of the particular language can perform the classification manually;
- they require high throughput because every message consists of many patterns;
- many (millions) of samples are available for training;
- formal models of *context* (morphological, lexical, syntactic, pragmatic [1]) and of *style* (typefaces, hand print, dialects [2, 3, 4]) have been developed;
- the error/reject tradeoffs are well understood [5];
- the classes are well defined: there are exactly ten digits and, in Italian, 21 letters of the alphabet;
- in feature space, the class centroids are located at the vertices of a regular simplex [6, 7, 8].

Examples of Type B include the recognition of birds, flowers and trees, many biometric applications, and biomedical pattern classification where the cost of preparing the samples often dominates the cost of recognition. Type B applications

- deal with natural patterns which may have developed without the discriminability of symbolic patterns;
- must be classified on demand rather than as part of a work-flow;
- are recognized only by relatively few, highly trained experts (bird-watchers, foresters, physicians);
- often have only small training sets because of the high cost of labeling;
- seldom have established models of context or style;
- because of the unpredictable cost of errors, require every decision to be checked by a human;
- exhibit a soft, hierarchical class structure, subject to change.

Type AB applications have some characteristics of both Type A and Type B. An example is genetic sequence decoding. We defer consideration of Type O.

3 Interaction

Almost all operational pattern recognition systems require some human interaction, at least at the beginning or end. We focus here on systems where human and machine take turns to reach a decision that assigns a particular object (a flower, tree, face, or skin lesion) to a particular class.

There are essential differences between human and machine cognitive abilities. Humans excel in gestalt tasks, like object-background separation. They apply to recognition a rich set of contextual constraints gleaned from previous experience rather than from specific training and test sets. They have superior noise-filtering abilities (particularly with respect to colored noise.) They can easily read degraded text on which the best OCR systems produce only gibberish. Computer vision systems, on the other hand, still have difficulty in recognizing "obvious" differences and "generalizing" from limited training sets.

Computers, however, can perform many tasks faster and better. Computers can store thousands of images and the associations between them, and never forget a name or a label. They can evaluate geometrical properties like high-order moments whereas a human is challenged to determine even the centroid of a convoluted figure. They can compute wavelets and other kernel transforms to differentiate textures. Computers can quickly measure lengths and areas, count thousands of connected components and sort them according to various criteria (size, aspect ratio, convexity). They can flawlessly evaluate multivariate conditional probabilities, decision functions, logic rules, and grammars. In contrast, George Miller's psychophysical experiments in the 1950's revealed that humans have limited short-term memory (only ±7 items) and poor absolute judgment.

We believe that the key to effective interactive recognition is a *visible model* to mediate human-computer communication. The model is a geometric and topological abstraction of an object represented in a picture. It guides the machine to extract

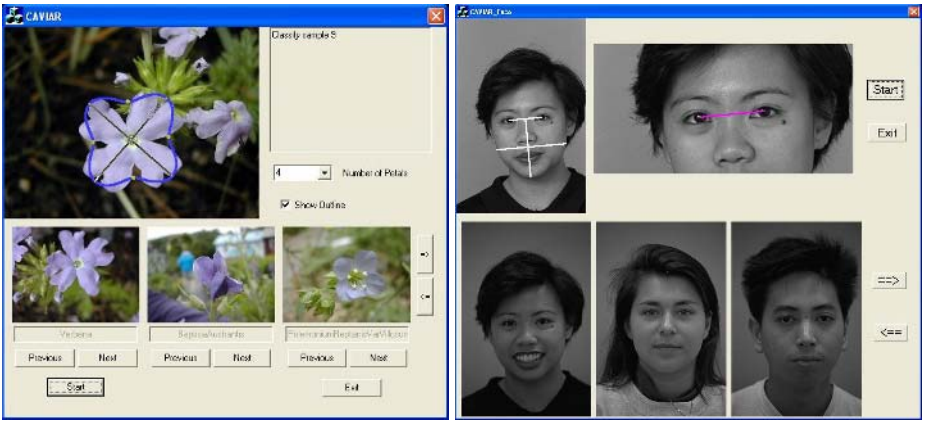


Fig. 1. CAVIAR-flower (left) and CAVIAR-face (right) visible models and graphic user interfaces. Because pupil localization is so important in face recognition, an enlarged view is provided for this part of the visible model.

discriminative, localized intensity, color and texture features. The model mediates only a restricted set of information. It does not tell the computer anything about the rich perceptions that lead the operator to correct or approve the model, and it does not tell the human about the configuration of the resulting feature vectors in high-dimensional feature space.

We have developed CAVIAR (*Computer Assisted Visual Interactive Recognition*) systems for some Type B applications [9 - 12]. Figure 1 shows examples of our flower and face models. These models are constructed automatically, and corrected interactively only when necessary. The line drawing superimposed on the picture lets the operator judge whether the computer-suggested model fits the unknown object. We restrict the model to isolated points and curves, because color and intensity are difficult to modify with a mouse or stylus. In effect, the user can point and drag, but not paint or shade. The model is only an abstraction: by itself, it is not sufficient for classification by either human or computer. Both must have unlimited access to, and make use of, the entire pixel array. The purpose of the model is only to guide the machine in extracting good features for classification.

A model instance need not depict faithfully intensity, color, or texture edges. A poorly fitting model may suffice to classify an "easy" object. Conversely, even an accurate model may result in ambiguous features. (One consequence of the role of the model in our system is that there can be no "ground truth" for it. Several models, or none, may lead to features that cause the correct candidate to be ranked on top.) The computer displays, in addition to the visible model, a set of reference pictures ranked according to the posterior class probabilities of the unknown object. The operator can correct the model if the top-ranked classes are implausible and if there are obvious mismatches between the current model and the unknown object. The operator can also scroll the reference patterns ("browse") to inspect candidates ranked below the top three. When an acceptable reference candidate appears on the display, the operator clicks on it to assign its class to the unknown object. Two CAVIAR graphic user interfaces (GUIs) are shown in Figure 1.

We compared CAVIAR to machine alone and to human alone in experiments conducted on a 612-flower database [10] and on the FERET face recognition benchmark [12]. Table 1 summarizes the results.

From this table, it appears that on some Type B applications interactive classification is more than twice as accurate than automated classification (at least with our classification algorithms), and more than twice as fast than unaided naïve human subjects.

Table 1. CAVIAR compared to machine alone and to human alone

	Flowers		Faces	
	Accuracy	Time per flower	Accuracy	Time per face
Interactive	93%	12 sec	99.7%	7.6 sec
Machine alone	32%	**---**	48.0%	----
Human alone	93%	26 sec	~100.0%	66.3 sec

In our CAVIAR-face experiments, 50 faces ("probes") were classified to one of 200 classes ("gallery") by naive subjects. Each subject classified a randomly selected set of 50 faces. The fraction of faces recognized correctly after each adjustment, and the time required, are shown in Table 2. For example, it takes 10.6 seconds to classify pictures that require two adjustments. Such pictures represent 15% of the total number of test pictures.

It is seen that the automated rank-ordering algorithm ranks the correct reference picture in the top three about 50% of the time. These faces are classified very quickly. Only about 1% of the faces require more than five model adjustments. Subjects seldom use the browsing option because it is slow. (A larger display would speed up the process further by displaying, with sufficient resolution for easy identification, more than the top three candidates.)

Table 2. Cumulative record of face recognition experiments. At each iteration, the operator can make an immediate decision (SELECT), modify the model (ADJUST), or look at more reference pictures (BROWSE). The times shown include the sequence of adjustments or browsing prior to classification. For example, after two adjustments, 85% of the faces are classified, in 5.0 seconds per face on average.

Iteration #	ADJUST %	SELECT %	Ave. time sec	BROWSE %	Ave. time sec	Classified %
0	48.7	50.3	2.3 sec	1.0	7.7	51.3
1	28.3	19.7	7.7 sec	0.7	16.1	71.7
2	15.0	13.3	10.6 sec	0.0	--	85.0
3	9.3	4.7	14.4 sec	1.0	42.6	90.7
4	3.7	5.3	16.6 sec	0.3	23.2	96.3
5	1.0	2.0	19.6 sec	0.7	33.2	99.0
6	0.7	0.3	42.0 sec	0.0	--	99.3
7	0.3	0.3	34.7 sec	0.0	--	99.6
8	0.0	0.0	--	0.4	49.8	100.0
Total		95.9		4.1		

The rank-ordering mechanism is chosen according to the number of classes and the number of reference samples per class. For CAVIAR-flower, where we ran experiments with 1, or 3, or 5 reference samples per class, we used nearest-neighbors with 8 features [10]. For CAVIAR-face, where we had only one reference sample per class, we used the Borda Count on small patch features near the eyes and mouth. The test samples are added to the reference set after they are classified, thereby gradually improving the statistical estimates for automated model construction and for rank ordering. Despite the occurrence of misidentified samples and the lack of adequate theoretical justification, we have always found this type of adaptive learning extremely effective [14 - 17, 10]

In summary, CAVIAR combines the following aspects that contribute to its accuracy and speed:

- interaction throughout the classification process, rather than only at the beginning or end;
- automatic correction and interactive modification of a simple, domain-specific visible model which guides automated feature extraction;
- pruning the reference pictures according to their similarity with the unknown, and displaying the top candidates for selection by the operator;
- adaptive re-estimation of internal parameters;
- leaving the human wholly in charge throughout the process by letting him or her decide when to classify an object, when to modify the model, and when to browse lower-ranked reference pictures.

4 Mobile Recognition

Mobile recognition systems with hand-held cameras offer obvious advantages for recognizing objects outside the office or home. The interaction takes place with stylus or thumb on the photo display screen. The computation can be carried out on the same platform or through a wireless link to a nearby laptop or to the Internet. Figure 2 shows our Mobile-CAVIAR interface on a Toshiba pocket computer with a plug-in camera and a Wi-Fi card [18].

One of the most interesting aspects of mobile systems is their potential capability to improve classification accuracy because the operator can immediately take additional pictures of a difficult object. Classification can then be based on several pictures by merging relevant information. However, we don't yet know just how to do this effectively. Note that this is neither classifier combination nor sensor fusion: it is more akin to 3-D model construction from multiple views, except that the objective here is classification rather than representation. Some of the problems awaiting solution are listed below [18].

- When is a single picture of the object not enough, and another required?
- What viewpoint, scale, or illumination is best for a second or third picture?
- Should the computer or the human decide? If the former, how should it specify the desired picture?
- At what level(s) (pixel, feature, classifier) should the information from multiple pictures be combined?
- Several views of an object would require these views to be represented in the reference set. What would be an appropriate sampling scheme to ensure this during enrollment of training/reference samples?

Although PDA cameras still lag stand-alone digital cameras in terms of optical and digital resolution (and convenience features), the greatest limitation of handheld systems compared to PC recognition platforms is their limited screen size, which prevents simultaneous display of several objects in adequate detail (multiple zooms are disorienting). It is clear that camera-phones will soon have enough storage and computing capacity, as well as appropriate operating systems, for interactive recognition. However, the display size limitation will be even more stringent – current cell phones have display sizes ranging from 96x64 to 128x160 pixels. Some expect that it will be overcome by wireless access to large public displays [19], but we can

hardly expect a public display to pop up whenever we wish to recognize a flower or a face. Furthermore, cell phones lack a direct pointing device: fitting a visible model with arrow keys is clumsy

Fig. 2. M-CAVIAR graphic user interface. The model does not have to be perfect to rank the unknown in the top 3

5 Pattern Recognition Networks

Interactive pattern recognition may be ready to benefit from the kind of distributed computing envisioned by the creators of the ARPANET 30 years ago. Experts anywhere on the Internet can, in principle, interact with any image through its visible model. Just as bird watchers and wildflower enthusiasts band together for collective judgment, dermatologists at dispersed locations will be able to pool their expertise to diagnose difficult skin lesions. The interaction may take place in parallel, or in a hierarchy where the more difficult pictures percolate to the more qualified experts.

As mentioned already, one advantage of human interaction is that newly labeled specimens can be safely added to the training or reference database, thus improving the performance of the automated model construction, feature extraction, and rank ordering. We don't yet know whether a democratic CAVIAR will be more accurate than a CAVIAR operated by a Designated Expert. In any case, the larger sample size resulting from many persons collecting pictures, correcting models, and assigning class labels should benefit overall performance [20].

A typical network interaction session of our wireless ethernet (802.11b) link through TCP/IP sockets is shown in Figure 3. Whenever the PDA sends data, it blocks execution flow till it receives a response from the server. After interpreting the required type of service according to the header data of the byte stream, the appropriate server routine is invoked. Each user interaction requires a response from the server to update the display, but the only long message is uploading a new picture to the host computer. With current GSM, TDMA or CDMA cell phone links, this would take several seconds even if the picture were first compressed. However, the third generation cellular networks already being deployed will have more than sufficient bandwidth.

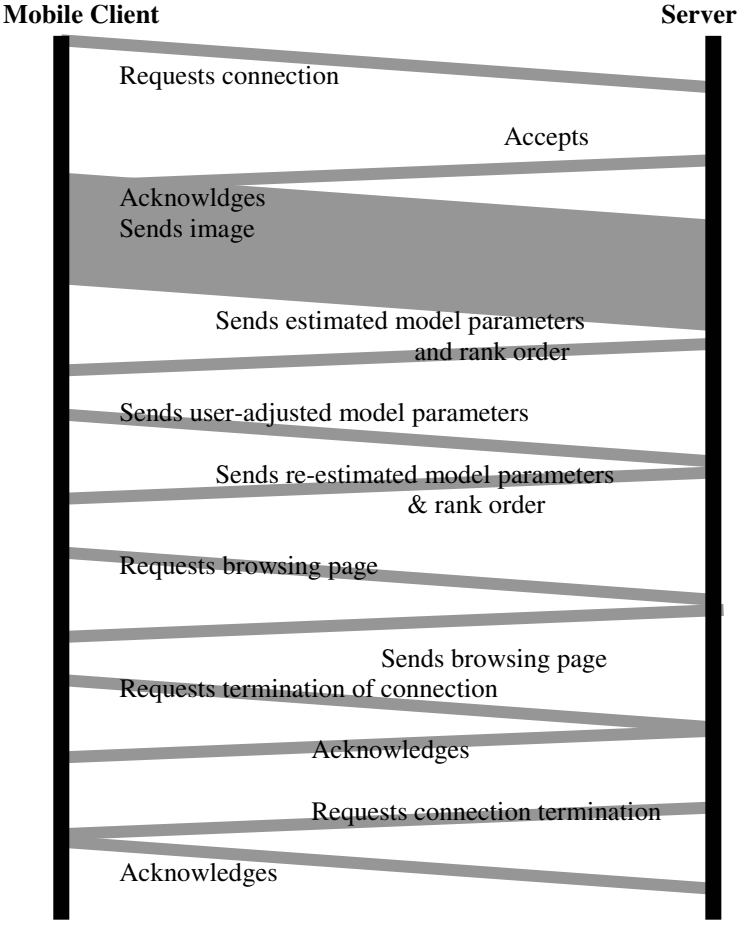


Fig. 3. Communication sequence between the PDA and the server for identifying a test sample {adapted from [18])

6 Style and Context

Interactive classification is too slow for Type A applications, so we must resort to other means to improve the conventional classification paradigm. Because they normally convey messages, symbolic patterns tend to appear together in groups (fields) that have a common origin. A printed message is usually a field of character patterns printed in the same font. Each font makes use of only a few well-matched typographic components—bowls, stems, bars, finials, and serifs. Printers, copiers and scanners leave individual imprints that differentiate documents from different sources. Hand printing and cursive writing are characterized by a certain writer-dependent uniformity of strokes and spacing. Therefore the feature-space representation of any single postal address, bureaucratic form, or printed article displays a measure of homogeneity due to isogeny (common origin). We say that isogeny induces *style* in features measured on patterns (cf. Fig 4a).

It is possible to model "style" mathematically and thereby develop a basis for more accurate classification of a group (field) of digitized characters from the same source. The features of patterns co-occurring in a field are statistically dependent because they share the same style. Effects of style consistency on the distributions of field-features (concatenation of pattern features) can be modeled by discrete, hierarchical, or continuous mixtures of Gaussian variables. Based on such a model, a style-constrained classifier can be constructed to recognize entire fields of patterns rendered in a consistent but unknown style. In experiments on printed alphabets and on National Institute of Standards and Technology (NIST) hand-printed test sets, style constrained classification reduced errors on fields of digits by nearly 25 percent over singlet classifiers [2, 3]. We are currently trying to develop style-constrained SVMs that should yield even lower error rates when the underlying densities are not Gaussian.

Fig. 4. (a) Style: in the part numbers on the left, the first pattern in the top row must be the letter l, and the identical first pattern in the bottom row must be the numeral 1. (b) Language context: here the surrounding text disambiguates the identical glyphs in either row.

Language context has been exploited in OCR and ASR far longer than style. Unlike style, textual context depends on the reading order of the patterns. Instead of modeling the dependence between the *features* of patterns in a same-source field, it models the dependence between their *labels* (Fig. 4b). For use in classification, morphological, lexical and syntactic conventions are converted into letter n-gram frequencies, word frequencies, and word transition probabilities. A vocabulary of 60,000-100,000 words provides very thorough coverage of English text, but three or four times more entries are needed for highly-inflected languages such as Italian or

Hungarian. Specialized lexicons are needed for mail sorting, telephone directories, technical handbooks, part-number catalogs, and other non-narrative compendia. Nowadays such information can be readily obtained from the Web.

Specialized syntactic conventions dictate the placement of punctuation marks and the construction of abbreviations, citations, legal and courtesy amounts on checks, and postal addresses. Mathematical formulas, chemical structure diagrams, and arithmetic redundancy in financial documents also have their own rules. These rules have been compiled in style guides ranging from a few pages for technical journals and mailing instructions to manuals of hundreds of (web) pages. Markov models provide an efficient and trainable alternative to formal parsing. Although several books have been published on the appropriate design and layout (i.e., syntax) of tables, tables have proved singularly resistant to correct interpretation.

In the last decade, segmentation, context, and shape based classification have been successfully integrated into algorithms that search a trellis of trial segmentations with transitions dictated by context. The weight given to shape information vs. contextual information depends on the application. In the limit, with no prior shape information but perfect shape consistency, the recognition problem is equivalent to decoding a substitution cipher [21 - 24].

The distinction between the various types of statistical dependence underlying correlated features, styles, and language context is clarified when they are modeled with graphical Bayesian networks [25].

7 Applications

Type A Applications. The manifold applications of character and speech recognition are already well known. Further research on style based recognition of fields of characters or even entire documents is much more convenient than on interactive recognition, because it does not require interaction with human subjects. We believe, however, that it too is inspired by human recognition. Indeed, the automatic recognition of *isolated* printed and hand-printed letters or digits is at least as accurate as human recognition. It is only when we are faced with a coherent sequence of patterns designed to deliver a message that human recognition is still far superior. Style and context can narrow the gap.

Type B applications. I believe that the time is right for developing interactive, mobile pattern recognition applications. I am excited about their potential impact on education, even though none of the new waves of technology – 16mm movies, radio, television, time-sharing, personal computers, laptops, web-teach – has lived up to its educational expectations. The necessary mobile wireless computing platforms (PDAs and cell phones) are already widespread among the school-age population. Education – from elementary to graduate school – involves many visual recognition tasks. At various stages of our academic careers, we learn to identify flowers, trees, rocks, insects, clouds, paintings, statues, architectural styles. Applications to industrial training – recognizing electrical and mechanical components and assemblies – may also be worth exploring. Interactive computer-aided recognition could enhance learning almost as much as visiting a zoological or botanical garden or a museum with an expert personal tutor. Group dynamics can be added by networked

recognition, either with nearby classmates or students and teachers at the antipodes. CAVIAR-like instructional systems will be considered successful only if, after a period of use, they can be discarded.

Visitors who saw a demonstration of CAVIAR have suggested several applications: searching for new medicinal plants in the jungle; identifying endangered cryptic cats (like jaguars) caught in photo-traps in Costa Rica; fruit and vegetable checkout in supermarkets; assisting farmers and foresters to identify pests and crop diseases; helping to learn the artistic style of painters in art history classes; assessing the value of collectibles (spoons, coins, stamps, porcelain dolls) using home web cams (to draw customers to advertising websites).

Among medical applications, the recognition of skin lesions seems particularly appropriate, because they are often diagnosed by inspection. Large dermatological atlases are available on the Web, but the most of the posted photographs were obtained at high resolution and with specialized lighting, and these databases contain only one or two examples of each disease. Collecting appropriate data with a mobile platform entails restrictive ethical considerations and will require close collaboration with medical researchers. Anyone of the following types of visible skin conditions appears ripe for experimentation:

- Cosmetic dermatology, scar assessment, beauty-aids;
- Infectious and contagious diseases with spots: measles, chickenpox, rubella;
- Rashes: hives, eczemas, psoriasis;
- Burns; cuts, frostbite;
- Sexually transmitted diseases;
- Poisonous plants and bugs: poison ivy, insect bites;
- Bio-terrorism agents: cutaneous anthrax, smallpox, plague, tularemia.

A personal diagnostic system may be appropriate when the patient is too embarrassed to seek help, or when medical personnel is not available (on battlefields or expeditions, or in impoverished rural areas). The same platform can help collect ancillary information in complete privacy: *Does it itch? How long have you had it? Do you feel lethargic? Did you eat fish recently?* Close-ups and pictures of healthy areas of the skin can be taken with different sources of illumination for comparison. If the severity of the condition warrants it, or if a confident diagnosis cannot be reached, the system may also request its owner to forward electronically both the pictures and the ancillary information to the appropriate health maintenance network. Paramedical personnel or physicians not specialized in dermatology could use CAVIAR-derma for continuing health care education.

Acknowledgments

Dr. Jie Zou (now at NIH) built the first CAVIAR. Hamei Jiang collected pictures of fruit, stamps, coins and Han characters for the early CAVIAR experiments. Greenie Cheng and Laura Derby photographed *many* flowers and helped build the database. Borjan Gagoski recruited subjects, conducted the 30 flower recognition experiments, and compiled the results. Rebecca Seth (City Naturalist, Lincoln, NE), Dr. Richard Mitchell (NY State Botanist) and Prof. Robert Ingalls (RPI CS Dept) gave us valuable

advice about the classification of plants and flowers. Arthur Evans, John Sikorski, and Patricia Thomas, under the supervision of Professors Sung-Hyuk Cha and Charles Tappert at Pace University, ported CAVIAR to the Zaurus. Abhishek Gattani (now with Aktina Medical Corporation) developed and tested M-CAVIAR as part of his MS thesis at Rensselaer. Prof. Qiang Ji (RPI ECSE Dept) suggested that we apply CAVIAR to face recognition and freely shared with us his data and insights. Drs Prateek Sarkar (now at PARC) and Harsha Veeramachaneni (now at IRST in Trento) developed the style concepts in the course of their doctoral research at RPI.

References

1. G. Nagy, Teaching a computer to read, *Procs. Int'l Conf. Pattern Recognition (ICPR XI),* vol. 2, pp. 225-229, The Hague, August 1992.
2. P. Sarkar, G. Nagy, Style consistent classification of isogenous patterns, *IEEE Trans. PAMI,* vol. 27, no. 1, pp. 88-98, Jan. 2005,
3. S. Veeramachaneni, G. Nagy, Style context with second order statistics, *IEEE Trans. PAMI,* vol. 27, no. 1, pp. 14-22, Jan. 2005.
4. N. Nagy, X. Zhang, G. Nagy, E.W. Schneider, A quantitative categorization of phonemic dialect features in context, *Procs. International Conference on Context,* Paris, July 2005.
5. G. Nagy, P. Sarkar, Document style census for OCR*, Procs. First International Workshop on Document Image Analysis for Libraries* (DIAL04), Palo Alto, CA, IEEE Computer Society Press, pp. 134-147, January 2004.
6. G. Nagy and S. Veeramachaneni, A Ptolemaic model for OCR, *Procs. ICDAR-03*, Edinburgh, pp. 1060-1064, August 2003.
7. G. Nagy, Visual pattern recognition in the years ahead, *Procs. ICPR XVII*, vol. IV, pp. 7-10, Cambridge, UK, August 2004.
8. G. Nagy, X. Zhang, Simple statistics for complex features spaces, to appear in Data Complexity in Pattern Recognition, Springer Verlag, Editors: Mitra Basu & Tin Kam Ho, Publication Date: December 2005.
9. G. Nagy, J. Zou, Interactive visual pattern recognition, *Procs. ICPR XVI,* IEEE Computer Society Press, Aug. 2002, Vol. III, pp. 478-481.
10. J. Zou, G. Nagy, Evaluation of model-based interactive pattern recognition, *Procs. ICPR XVII*, Vol. II, p. 311-314, Cambridge, UK, August 2004.
11. J. Zou, G. Nagy, Human-computer interaction for complex pattern recognition problems, to appear in Data Complexity in Pattern Recognition, Springer Verlag, Editors: Mitra Basu & Tin Kam Ho, December 2005.
12. A. Evans, J. Sikorski, P. Thomas, S-H Cha, C. Tappert, J. Zou, A. Gattani, G. Nagy, Computer Assisted Visual Interactive Recognition (CAVIAR) Technology, *Procs. IEEE International Conference on Electro-Information Technology,* Lincoln, NE May 2005.
13. J. Zou, A Model-Based Interactive Image Segmentation Procedure, *IEEE Workshop on Applications of Computer Vision (WACV), Breckenridge CO, January* 2005.
14. G. Nagy, G.L. Shelton, Self-Corrective Character Recognition System, *IEEE Transactions on Information Theory IT-12,* #2, pp. 215-222, April 1966.
15. H.S. Baird, G. Nagy, A Self-correcting 100-font Classifier, *Proc. SPIE Conference on Document Recognition, Volume SPIE-2181,* pp. 106-115, San Jose, CA, February 1994.
16. S. Veeramachaneni, G. Nagy, Adaptive classifiers for multi-source OCR, *IJDAR* vol 6*,* no. 3, pp. 154-166, March 2004.

17. G. Nagy, Classifiers that improve with use, *Procs. Conference on Pattern Recognition and Multimedia ,* IEICE, Tokyo, pp. 79-86, February 2004.
18. A. Gattani, Mobile Interactive Visual Pattern Recognition, MS thesis, Rensselaer Polytechnic Institute, December 2004.
19. M. Raghunath, C. Narayanaswami, and C. Pinhanez, Fostering a Symbiotic Handheld Environment, *IEEE Computer,* pp. 56-65, Sept. 2004.
20. J. Zou, A. Gattani, Computer Assisted Visual InterActive Recognition and Its Prospects of Implementation Over the Internet, *IS&T/SPIE 17th Annual Symposium Electronic Imaging, Internet Imaging VI*, 2005.
21. R.G. Casey, G. Nagy, Autonomous Reading Machine, *IEEE Transactions on Computers C-17*, #5, pp. 492-503, May 1968.
22. R.G. Casey, G. Nagy, Advances in Pattern Recognition, *Scientific American 224,* #4, pp. 56-71, 1971.
23. G. Nagy, S. Seth, K. Einspahr, Decoding Substitution Ciphers by means of Word Matching with Application to OCR, *IEEE Transactions on Pattern Analysis and Machine Intelligence 9,* #5, pp. 710-715, September 1987.
24. T.K. Ho, G. Nagy, OCR with no shape training, *Procs. ICPR-XV,* vol. 4, pp. 27-30, Barcelona, September 2000.
25. S. Veeramachaneni, P. Sarkar, G. Nagy, Modeling context as statistical dependence, *Procs. International Conference on Context,* Paris, July 2005.

Most of these papers are available as PDF files at: http://www.ecse.rpi.edu/homepages/nagy/

Learning Intrusion Detection: Supervised or Unsupervised?

Pavel Laskov, Patrick Düssel, Christin Schäfer, and Konrad Rieck

Fraunhofer-FIRST.IDA, Kekuléstr. 7, 12489 Berlin, Germany
{laskov, duessel, christin, rieck}@first.fhg.de

Abstract. Application and development of specialized machine learning techniques is gaining increasing attention in the intrusion detection community. A variety of learning techniques proposed for different intrusion detection problems can be roughly classified into two broad categories: supervised (classification) and unsupervised (anomaly detection and clustering). In this contribution we develop an experimental framework for comparative analysis of both kinds of learning techniques. In our framework we cast unsupervised techniques into a special case of classification, for which training and model selection can be performed by means of ROC analysis. We then investigate both kinds of learning techniques with respect to their detection accuracy and ability to detect unknown attacks.

1 Introduction

Intrusion detection techniques are usually classified into misuse detection and anomaly detection [1]. Anomaly detection focuses on detecting unusual activity patterns in the observed data [2,3,4,5,6]. Misuse detection methods are intended to recognize known attack patterns. Signature-based misuse detection techniques are currently most widely used in practice; however, interest is growing in the intrusion detection community to application of advances machine learning techniques [7,8,9,10]. Not uncommon is also a combination of anomaly and misuse detection in a single intrusion detection system.

To decide which learning technique(s) is to be applied for a particular intrusion detection system, it is important to understand the role the label information plays in such applications. The following observations should be considered:

1. Labels can be extremely difficult or impossible to obtain. Analysis of network traffic or audit logs is very time-consuming and usually only a small portion of the available data can be labeled. Furthermore, in certain cases, for example at a packet level, it may be impossible to unambiguously assign a label to a data instance.
2. In a real application, one can never be sure that a set of available labeled examples covers all possible attacks. If a new attack appears, examples of it may not have been seen in training data.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 50–57, 2005.

The main goal of this work is to investigate the tradeoffs between supervised and unsupervised techniques in their application to intrusion detection systems. To this end, we develop an experimental setup in which such techniques can be fairly compared. Our setup is based on the well-known KDD Cup 1999 data set [11]. Although this data set is known to have certain drawbacks, caused by the artificial nature of underlying data in DARPA IDS evaluations, no other data sets are currently available for comprehensive experimental studies. Since a typical application of a supervised learning method involves model selection, we have built in model selection into unsupervised methods. Performance of both groups of methods is evaluated based on the analysis of the receiver operator characteristic (ROC) curve. The details of our experimental setup are presented in Sec. 2.

Evaluation of several representative supervised and unsupervised learning algorithms, briefly reviewed in Sec. 3, is carried out under the following two scenarios. Under the first scenario, an assumption that training and test data come from the same (unknown) distribution is fulfilled. Under the second scenario, we violate this assumption by taking a data set in which attacks unseen in training data are present in test data. This is a typical scheme to test the ability of an IDS to cope with unknown attacks. The experimental results are presented in Sec. 4.

2 Experimental Setup

2.1 Data Source

The KDD Cup 1999 data set [11] is a common benchmark for evaluation of intrusion detection techniques. It comprises a fixed set of connection-based features. The majority of instances in this set (94%, 4898430 instances) has been extracted from the DARPA 1998 IDS evaluation [12]. The remaining fraction of data (6%, 311029 instances) was additionally extracted from the extended DARPA 1999 IDS evaluation [13]. A detailed description of the available features and attack instances can be found in [14,6].

2.2 Preprocessing

The KDD Cup data set suffers from two major flaws in distribution of data which can bias comparative experiments:

1. The attack rate within the KDD Cup data set is unnatural. About 80% of all instances correspond to attacks, since all one-packet attacks, e.g. the `smurf` attack, are treated as full-value connections and are represented as individual instances.
2. The attack distribution within the KDD Cup data set is highly unbalanced. It is dominated by probes and denial-of-service attacks, which cover millions of instances. The most interesting and dangerous attacks, e.g. the `phf` or `imap` attacks, are grossly under-represented.

Table 1. Distribution of attack types in the experiments

"Known" Attack Types	"Unknown" Attack Types
back buffer_overflow ftp_write guess_passwd imap ipsweep land loadmodule multihop neptune nmap perl phf pod portsweep rootkit satan smurf spy teardrop warezclient warezmaster	apache2 httptunnel mailbomb mscan named processtable ps saint sendmail snmpgetattack snmpguess sqlattack updstorm worm xlock xsnoop xterm

In order to cope with these artifacts we preprocess KDD Cup data in order to achieve (a) a fixed attack rate and (b) a balanced distribution of attack and service types.

At the first level of preprocessing, the attack data is split into disjoint partitions containing only one attack type. The normal data is split into disjoint partitions containing only one service type. These partitions are merged into the three disjoint sets of equal length: the training data D_{train}, the validation data D_{val} and the test data D_{test}. This procedure ensures the presence of each attack and service type in the three data partitions.

At the second level of preprocessing samples of 2000 instances are randomly drawn from the training, validation and testing data sets. The sampling procedure enforces a fixed attack rate of 5% and attempts to preserve balanced attack and service type distributions.

The data with "known attacks" is generated from the DARPA 1998 part of the KDD Cup data set. The data with "unknown attacks" has the test part sampled from the DARPA 1999 part of the KDD Cup data set. The attacks in both data sets are listed in Table 1.

2.3 Metric Embedding

The set of features present in the KDD Cup data set contains categorical and numerical features of different sources and scales. An essential step for handling such data is *metric embedding* which transforms the data into a metric space. Our embedding is a two-stage procedure similar to [3,2].

Embedding of Categorical Features. Each categorical feature expressing m possible categorical values is transformed to a value in $\mathbb{R}^m$ using a function e that maps the j-th value of the feature to the j-th component of an m-dimensional vector:

$$e(x_i) = \underbrace{(0, \ldots, 1, \ldots, 0)}_{1 \text{ at Position } j} \quad \text{if } x_i \text{ equals value } j$$

Scaling of Features. Both the numerical and the embedded categorical features are scaled with respect to each feature's mean μ and standard deviation σ:

$$n(x_i) = \frac{x_i - \mu}{\sigma}$$

2.4 Model Selection

Model selection is performed by training a supervised algorithm on a training set D_{train} and evaluating the accuracy on 10 validation sets D_{val} generated as described in Sec. 2. For unsupervised algorithms only evaluation is performed. The criterion for evaluating the accuracy is the area under the ROC curve, computed for the false-positive interval [0, 0.1].

3 Methods

In the following we briefly describe the algorithms used in our experiments.

3.1 Supervised Algorithms

C4.5. The C4.5 algorithm [15] performs inference of decision trees using a set of conditions over the attributes. Classification of new examples is carried out by applying the inferred rules. Although the original algorithms contains numerous free parameters, only the number of bootstrap iterations was used in our evaluation.

***k*-Nearest Neighbor.** The k-Nearest Neighbor is a classical algorithm (e.g. [16]) that finds k examples in training data that are closest to the test example and assigns the most frequent label among these examples to the new example. The only free parameter is the size k of the neighborhood.

Multi-layer Perceptron. Training of a multi-layer perceptron involves optimizing the weights for the activation function of neurons organized in a network architecture. The global objective function is minimized using the RPROP algorithm (e.g. [17]). The free parameter is the number of hidden neurons.

Regularized Discriminant Analysis. Assuming both classes of examples are normally distributed, a Bayes-optimal separating surface is a hyperplane (LDA), if covariance matrices are the same, or a quadratic surface otherwise (QDA). A gradual morph between the two cases can be implemented by using a regularization parameter γ [18]. Another free parameter λ controls the addition of identity matrix to covariance matrices.

Fisher Linear Discriminant. Fisher Linear Discriminant constructs a separating hyperplane using a direction that maximizes inter-class variance and minimized the intra-class variance for the projection of the training points on this direction (e.g. [16]). The free parameter is the tradeoff between the norm of the direction and the "strictness" of projection.

Linear Programming Machine and Support Vector Machine. Linear Programming Machine (LPM) and Support Vector Machine (SVM) construct a hyperplane of the minimal norm which separates the two classes of training examples (e.g. [19]). LPM uses the 1-norm, SVM uses the 2-norm. Furthermore,

SVM apply a non-linear mapping to construct a hyperplane in a feature space. In our experiments, radial basis functions are used, their complexity controlled by the width parameter w. Another parameter C controls the tradeoff between the norm of a hyperplane and the separation accuracy.

3.2 Unsupervised Algorithms

γ-Algorithm. The γ-algorithm is a recently proposed graph-based outlier detection algorithm [20]. It assigns to every example the γ-score which is the mean distance to the example's k nearest neighbors. The free parameter is k.

***k*-Means Clustering.** k-Means clustering is a classical clustering algorithm (e.g. [16]). After an initial random assignment of example to k clusters, the centers of clusters are computed and the examples are assigned to the clusters with the closest centers. The process is repeated until the cluster centers do not significantly change. Once the cluster assignment is fixed, the mean distance of an example to cluster centers is used as the score. The free parameter is k.

Single Linkage Clustering. Single linkage clustering [2] is similar to k-Means clustering except that the number of clusters is controlled by the distance parameter W: if the distance from an example to the nearest cluster center exceeds W a new cluster is set.

Quarter-Sphere Support Vector Machine. The quarter-sphere SVM [5,6] is an anomaly detection method based on the idea of fitting a sphere onto the center of mass of data. An anomaly score is defined by the distance of a data point from the center of the sphere. Choosing a threshold for the attack scores determines the radius of the sphere enclosing normal data points.

4 Results

The supervised and the unsupervised algorithms are evaluated separately on the data with known and unknown attacks. The results are shown in Figs. 1 and 2 respectively. The ROC curves are averaged over 30 runs of each algorithm by fixing a set of false-positive rate values of interest and computing the means and the standard deviations of true-positive rate values over all runs for the values of interest.

The supervised algorithms in general exhibit excellent classification accuracy on the data with known attacks. The best results have been achieved by the C4.5 algorithm which attains the 95% true positive rate at 1% false-positive rate. The next two best algorithms are the MLP and the SVM, both non-linear, followed by the local k-Nearest Neighbor algorithm. The difference between the four best methods is marginal. The worst results were observed with the three linear algorithms. One can thus conclude that a decision boundary between the attack and the normal data in KDD Cup features is non-linear, and is best learned by non-linear algorithms or their approximations.

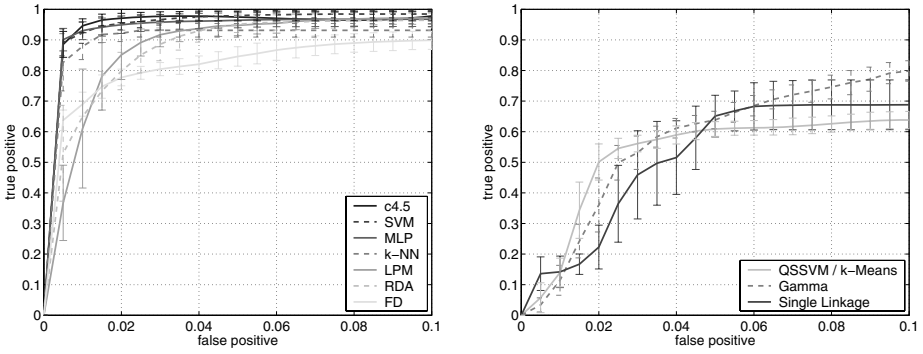


Fig. 1. ROC-curves obtained on *known* attacks: supervised (left) and unsupervised (right) methods

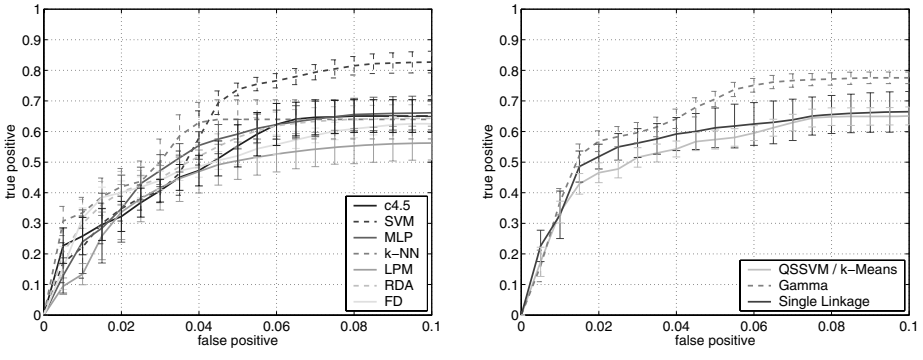


Fig. 2. ROC-curves obtained on *unknown* attacks: supervised (left) and unsupervised (right) methods

The accuracy of supervised algorithms deteriorates significantly if unknown attacks are present in the test data, as can be seen in the left part of Fig. 2. Not all algorithms generalize equally well to the data with unknown attacks. The best results (with a significant margin) are attained by the SVM, which can be attributed to the fact that the free parameters of this algorithm are motivated by learning-theoretic arguments aimed at maintaining an ability to generalize to unseen data. The next best contestant is the k-Nearest Neighbor algorithm which possesses the most similarity to the unsupervised methods. The remaining algorithm perform approximately equally.

The unsupervised algorithms exhibit no significant difference in performance between known and unknown attacks. This result is not unexpected: in fact, in all data sets the attacks are unknown to the algorithms – the two data sets differ merely in the set of attacks contained in them. Among the algorithms the preference should be given to the γ-algorithm which performs especially well on the "unknown" data set. The accuracy of unsupervised algorithms on both

data sets is approximately the same as that of supervised algorithms on the "unknown" data set.

5 Conclusions

We have presented an experimental framework in which supervised and unsupervised learning methods can be evaluated in an intrusion detection application. Our experiments demonstrate that the supervised learning methods significantly outperform the unsupervised ones if the test data contains no unknown attacks. Furthermore, among the supervised methods, the best performance is achieved by the non-linear methods, such as SVM, multi-layer perceptrons, and the rule-based methods. In the presence of unknown attacks in the test data, the performance of all supervised methods drops significantly, SVM being the most robust to the presence of unknown attacks.

The performance of unsupervised learning is not affected by unknown attacks and is similar to the performance of the supervised learning under this scenario. This makes the unsupervised methods, which do not require a laborious labelling process, a clear forerunner for practical purposes if unknown attacks can be expected.

Our findings suggest that the problem of test data being drawn from a different distribution cannot be solved within the purely supervised or unsupervised techniques. An emerging field of semi-supervised learning offers a promising direction of future research.

Acknowledgements. The authors gratefully acknowledge the funding from *Bundesministerium für Bildung und Forschung* under the project MIND (FKZ 01-SC40A), We would like to thank Stefan Harmeling and Klaus-Robert Müller for fruitful discussions and valuable help in the implementation of the algorithms.

References

1. Bace, R., Mell, P.: NIST special publication on intrusion detection systems. National Institute of Standards and Technology (2001)
2. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: Proc. ACM CSS Workshop on Data Mining Applied to Security. (2001)
3. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. In: Applications of Data Mining in Computer Security. Kluwer (2002)
4. Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., Srivastava, J.: A comparative study of anomaly detection schemes in network intrusion detection,. In: Proc. SIAM Conf. Data Mining. (2003)
5. Laskov, P., Schäfer, C., Kotenko, I.: Intrusion detection in unlabeled data with quarter-sphere support vector machines. In: Proc. DIMVA. (2004) 71–82
6. Laskov, P., Schäfer, C., Kotenko, I., Müller, K.R.: Intrusion detection in unlabeled data with quarter-sphere support vector machines (extended version). Praxis der Informationsverarbeitung und Kommunikation **27** (2004) 228–236

7. Ghosh, A.K., Schwartzbard, A., Schatz, M.: Learning program behavior profiles for intrusion detection. In: Proc. of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, USA (1999) 51–62 http://www.cigital.com/papers/download/usenix_id99.pdf.
8. Warrender, C., Forrest, S., Perlmutter, B.: Detecting intrusions using system calls: alternative data methods. In: Proc. IEEE Symposium on Security and Privacy. (1999) 133–145
9. Mukkamala, S., Janoski, G., Sung, A.: Intrusion detection using neural networks and support vector machines. In: Proceedings of IEEE Internation Joint Conference on Neural Networks. (2002) 1702–1707
10. Lee, W., Stolfo, S., Mok, K.: A data mining framework for building intrusion detection models. In: Proc. IEEE Symposium on Security and Privacy. (1999) 120–132
11. Stolfo, S.J., Wei, F., Lee, W., Prodromidis, A., Chan, P.K.: KDD Cup - knowledge discovery and data mining competition (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
12. Lippmann, R., Cunningham, R.K., Fried, D.J., Kendall, K.R., Webster, S.E., Zissman, M.A.: Results of the DARPA 1998 offline intrusion detection evaluation. In: Proc. RAID 1999. (1999). http://www.ll.mit.edu/IST/ideval/pubs/1999/RAID_1999a.pdf.
13. Lippmann, R., Haines, J.W., Fried, D.J., Korba, J., Das, K.: The 1999 DARPA off-line intrusion detection evaluation. Computer Networks **34** (2000) 579–595
14. Lee, W., Stolfo, S.: A framework for constructing features and models for intrusion detection systems. In: ACM Transactions on Information and System Security. Volume 3. (2001) 227–261
15. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1992)
16. Duda, R., P.E.Hart, D.G.Stork: Pattern classification. second edn. John Wiley & Sons (2001)
17. Rojas, R.: Neural Networks: A Systematic Approach. Springer-Verlag, Berlin, Deutschland (1996)
18. Friedman, J.: Regularized discriminant analysis. Journal of the American Statistical Association **84** (1989) 165–175
19. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge, MA (2002)
20. Harmeling, S., Dornhege, G., Tax, D., Meinecke, F., Müller, K.R.: From outliers to prototypes: ordering data. Unpublished manuscript (http://ida.first.fhg.de/∼harmeli/ordering.pdf), submitted. (2004)

Network Intrusion Detection by Combining One-Class Classifiers

Giorgio Giacinto, Roberto Perdisci, and Fabio Roli

Department of Electrical and Electronic Engineering,
University of Cagliari, Piazza d'Armi - 09123 Cagliari, Italy
{giacinto, roberto.perdisci, roli}@diee.unica.it

Abstract. Intrusion Detection Systems (IDSs) play an essential role in today's network security infrastructures. Their main aim is in finding out traces of intrusion attempts alerting the network administrator as soon as possible, so that she can take suitable countermeasures. In this paper we propose a *misuse-based* Network Intrusion Detection architecture in which we combine multiple one-class classifiers. Each one-class classifier is trained in order to discriminate between a specific attack and all other traffic patterns. As attacks can be grouped in classes according to a taxonomy, for each attack class a number of one-class classifiers are trained, each one specialized to a specific attack. The proposed multiple classifier architecture combine the outputs of one class classifiers to attain an IDS based on *generalized attack signatures*. The aim is in labelling a pattern either as normal or as belonging to one of the attack classes according to the adopted taxonomy. The potentials and effectiveness of the proposed approach are analysed and discussed.

Keywords: Computer Security, Pattern Recognition.

1 Introduction

Intrusion Detection Systems (IDSs) are an important component of a defence-in-depth network security infrastructure. IDSs collect and analyse audit data looking for anomalous or intrusive activities. As soon as a suspicious event is detected an alarm is raised, so that the network administrator can react by applying suitable countermeasures. With respect to the source of the collected data, IDSs are divided into host-based and network-based. Host-based detectors collect audit data from operating systems facilities, application logs, file system information, etc., whereas network-based detectors collect data from packets crossing a network segment. IDSs can be further subdivided in two categories with respect to the implemented detection technique, namely misuse-based, and anomaly-based IDSs. If we view the intrusion detection problem as an instance of the generic signal-detection problem, we can consider the attacks as the signal to be detected and the normal activities as the noise [1]. Misuse-based (a.k.a. signature-based) detectors base their decisions on signal characterization, whereas anomaly-based detectors base their decisions on noise characterization. Accordingly, in order to detect an attack, a misuse-based IDS must possess a description of the attack

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 58–65, 2005.

which can be matched to the attack manifestations (i.e. the signal). Such a description is often called *attack signature*. Conversely, anomaly-based detectors rely on the assumption that attack manifestations are somehow distinguishable from the users' normal activities (i.e. the noise). Therefore, for an anomaly-based IDS to detect an attack, it must possess a model of the users' normal behaviour profile which can be compared to the attack manifestations. Despite research on the Intrusion Detection field has been active since 1980s, a number of questions about their effectiveness still remain unanswered [2]. This is mainly due to the difficulties in constructing both attack signatures and normal behaviour effective models. The misuse-based approach allows the detection of intrusions whose manifestations perfectly match the related attack signature, so that unknown (i.e. never seen before) intrusions can unlikely be detected. This problem could be solved by designing general signatures which should allow to detect at least most attack variants. Unfortunately, this is usually a hard designing task. Besides, implementing general signatures usually makes misuse-based IDSs prone to false alarms. On the other hand, anomaly-based approaches should allow the detection of both known and unknown attacks. However, due to the difficulties in choosing suitable models to characterize the normal users' activities, anomaly-based approaches usually produce a higher false alarm rate than misuse-based approaches.

It is straightforward that the trade-off between the ability to detect new attacks and the ability to generate a low false alarm rate is the key point to develop an effective IDS. Due to its ability to produce low false alarm rates, the misuse-based (signature-based) detection is currently the most widely used in enterprise environments, even at the price of a very limited ability to detect unknown attacks. In order to overcome the difficulties in detecting new attacks, a number of researchers have applied statistical pattern recognition approaches [3][4][5]. The main motivation in using pattern recognition approaches to develop advanced IDSs is their generalization ability, which may support the recognition of previously unseen intrusions that have no previously described patterns. As stated before, signature generalization could make the detector prone to false alarms. In this paper we propose a network-based Intrusion Detection System (NIDS) implemented by combining several one-class classifiers (i.e. classifiers trained using example of only one class) in order to achieve the detection of unknown attacks while keeping the false alarm rate as low as possible. Each one-class classifier is trained in order to discriminate between a specific attack and all other traffic patterns (the so called outliers). As attacks can be grouped in classes according to a taxonomy (e.g. the Weber-Kendall taxonomy [6][7]), for each attack class a number of one-class classifiers are trained, each one specialized to a specific attack. The proposed multiple classifier architecture combines the outputs of one class classifiers to attain an IDS based on generalized attack signatures. The aim is in labelling a pattern either as normal or as belonging to one of the attack classes according to the adopted taxonomy, keeping the detection rate as high as possible, the false positive rate as low as possible and recognizing as many new attacks as possible.

The paper is organised as follows. A formulation of the intrusion detection problem as an instance of the pattern classification problem is presented in Section 2. The one-class classifiers ensemble architecture is illustrated in section 3. Section 4 outlines the method used to fuse the outputs of the different one-class classifiers. The experimental evaluation of the proposed architecture using the KDD Cup 1999 data set, distributed as part of the UCI KDD Archive [8], is reported in Section 5. Conclusions are drawn in Section 6.

2 Problem Formulation

From the point of view of pattern recognition, the network intrusion detection problem can be formulated as follows: given the information about network connections between pairs of hosts, assign each connection to one out of N data classes representing normal traffic or different categories of intrusions. Computer attacks can be carried out using several techniques, each one trying to exploit operating system or application flaws. Several attack taxonomies have been proposed in the literature. An attack taxonomy aims at grouping the attacks according to a given criteria. For example, the Weber-Kendal taxonomy [6][7] groups the attacks in four classes, each class containing attacks which share the final goal, even if distinct attacks in a group are carried out using different techniques. The term connection refers to a sequence of data packets related to a particular service, e.g., the transfer of a web page via the http protocol. As the aim of a network intrusion detector is to detect connections related to malicious activities, each network connection can be defined as a pattern to be classified. Extraction of suitable features representing network connections is based on expert knowledge about the characteristics that distinguish attacks from normal connections. These features can be subdivided into two groups: features related to the data portion of packets (called payload) and features related to the network characteristics of the connection, extracted from the TCP/IP headers of packets [9]. The latter group of features can be further subdivided into two groups: intrinsic features, i.e., characteristics related to the current connection, and traffic features, related to a number of similar connections. Therefore, the following three feature sets can be used to classify each connection [10]:

- *content features*, i.e., features containing information about the data content of packets (payload) that could be relevant to discover an intrusion, e.g., errors reported by the operating system, root access attempts, etc.
- network related features
  - *intrinsic features*, i.e., general information related to the connection. They include the duration, type, protocol, flags, etc. of the connection;
  - traffic features, i.e., statistics related to past connections similar to the current one e.g., number of connections with the same destination host or connections related to the same service in a given time window or within a predefined number of past connections.

This feature categorisation is general enough to take into account the high number of features that can be used to describe network traffic.

3 A Multiple One-Class Classifier System

3.1 One-Class Classification

In conventional supervised classification problems we have example patterns from each of the classes of objects we would like to recognize. For example, in a two-classes classification problem the training set is made of a number of patterns from class A as well as a number of patterns from class B. Learning data labelled either as A or B are usually equally balanced. There exist problems for which it is difficult to produce example patterns from both the two classes. In such problems we need classifiers which are able to learn in absence of counter-examples [11]. Therefore, a one-class classifier is trained using examples from only the target class A. During operational use, the classifier aims at discriminating between patterns to be labelled as A and patterns not belonging to class A (the so called outliers). In the literature a number of different terms have been used for this problem, among them the most usual are one-class classification, novelty detection, outlier detection.

3.2 The Proposed Multiple One-Class Classifier Architecture

Let us assume that a training set containing connection examples from N different attacks as well as from normal activities is available. As stated in section 2, attacks can be grouped according to a taxonomy, therefore connection examples related to the attacks can be grouped according to the chosen taxonomy as well. Assuming that the taxonomy groups the attacks in M classes, the proposed architecture is made up of N one-class classifiers grouped in M groups. Each classifier is trained using all the training patterns related to an attack j from a given attack group i. Therefore, classifier c_{ij} has to discriminate between patterns related to attack (i, j) and patterns related to all other traffic patterns. For each connection pattern, N output values are produced (i.e. one output for each classifier) which are fused according to a decision function $f()$. Fusing the outputs produced by the classifiers of the i-th group allows attaining a generalized signature for the attack class i. For each pattern, the ensemble overall output will be a label indicating whether the pattern is related either to normal activities or to one among the M attack classes. Details about the implemented decision function will be explained in section 4.

4 Decision Function

For each pattern processed by the classifier ensemble, an output vector $\mathbf{o}$ is produced. Each entry o_h of the vector represents the output O_{ij} of one among the N classifiers c_{ij} (Figure 1). As an example, let us assume that attack class l is represented by k_l distinct attack types in the training set. Thus, the elements of the output vector from position v ($v = 1 + \sum_i k_i$, $i = 1, .., l - 1$) to position $v + k_l$ are produced by the set of one-class classifiers c_{l1}, c_{l2},..., c_{lk_l}, i.e. those

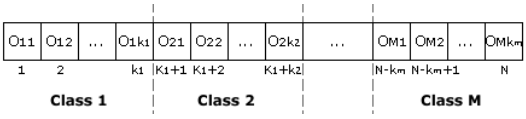


Fig. 1. Output vector

classifiers which are in charge of recognizing attacks belonging to the l-th class. The basic decision rule is straightforward. For a given test pattern, let us define

$$p = arg \max_{h=1..n} \{o_h\} \tag{1}$$

$$m = \max_{h=1..n} \{o_h\} \tag{2}$$

If $m < n_{th}$, where n_{th} is a predetermined threshold, then the pattern will be labelled as normal regardless of the position p. If $m > n_{th}$, then the position p determines the attack class the pattern belongs to. In particular, if $(v \leq p \leq v + k_l)$ then the pattern is labelled as belonging to attack class l. The threshold n_{th} is estimated by the decision template of the proposed multiple classifier system [12]. We defined the decision template as a matrix where each cell (r, s) contains the average output x of classifier c_{lj} over all the training patterns drawn from an attack class r, where the column index $s = j + \sum_i k_i$, $i = 1, .., l - 1$.

The threshold n_{th} can be tuned so that few attacks are incorrectly labeled as normal traffic. A number of additional decision thresholds can also be introduced in account of misclassification among different attack classes, and normal traffic being labeled as attacks. For example, if r refers to normal traffic and s refers to classifier c_{lj} , the value x in the cell (r, s) is interpreted as the average output of classifier c_{lj} for all the training patterns of normal traffic. If $x > n_{th}$, then the basic decision rule will probably assign a number of normal patterns to class l. To avoid this problem, we can set a new threshold $n_{lj_{th}} > x$ that will be taken into account only if m is greater than n_{th} and the position p identifies the output of the classifier c_{lj}.

5 Experimental Result

Experiments were carried out on the KDD Cup 1999 dataset distributed as part of the UCI KDD Archive [8]. The dataset is made up of a large number of network connections related to simulated normal and malicious traffic. Each connection is represented with a 41-dimensional feature vector according to the set of features illustrated in section 2. In particular, 9 features were of the intrinsic type, 13 features were of the content type, and the remaining 19 features were of the traffic type. Connections are also labelled as belonging to one out of five classes, i.e., normal traffic,Denial of Service (DoS) attacks, Remote to Local (R2L) attacks, User to Root (U2R) attacks, and Probing attacks according to the Weber-Kendal taxonomy [6][7]. Each attack class is made up of different attacks designed to attain the same effect by exploiting different vulnerabilities of the

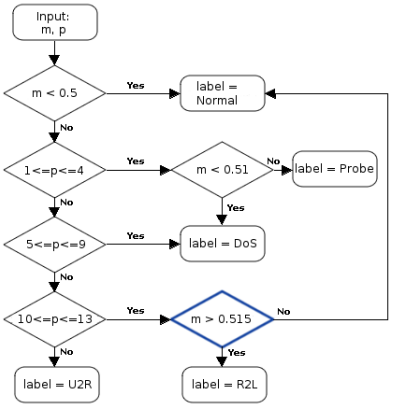


Fig. 2. Decision *schema 1* (adding "AND $N > 6$" into the highlighted rhomb we obtain the *schema 2*)

computer network. The original training set was made up of 492017 patterns related to different attacks belonging to one of the four attack categories. As the number of patterns related to the Smurf and Neptune attacks (DoS type) was very large compared to the number of patterns related to other attacks, we generated a new training set with balanced classes by pruning the patterns related to these attacks. The obtained training set contained 107701 patterns. The test set was made up of 311029 patterns, where 19091 patterns were related to attacks which belonged to one of the 4 attack classes (Probe, DoS, R2L, U2R) but which were not included in the training set. These patterns allow testing the ability of the pattern recognition approach to detect novel attack types (i.e. attacks not present in the training set). The training portion of the KDD Cup 1999 dataset contains 21 different attacks. According to the ensemble architecture described in sections 3 and 4, we have first trained 21 one-class classifiers using the k-means algorithm [11] setting k=1 and a reject percentage equal to 5%. The threshold n_{th} (section 4) has been set equal to 0.5. Each classifier has been trained using patterns related to a different attack from the training set. Afterwards, the decision template has been computed. Some of the 21 classifiers exhibited an average output grater than n_{th} for a large number of patterns related either to normal activities or to different attack classes. These classifiers were disregarded thus reducing the number of classifiers from 21 to 14 (table 1). It is worth noting that the 7 excluded classifiers were those which resulted to be heavily undertrained because of the very limited number of available training patterns. Nevertheless, the 7 attack types related to the excluded classifers may be still detected and classified as belonging to one out of the 4 attack classes thanks to the generalizion ability of the ensamble.

Performances have been first computed according to the basic decision rule described in section 4 with $n_{th} = 0.5$. Analysing the decision template we noted that some classifiers in the R2L group produced an average output greater than n_{th} when normal patterns were given as input. Besides, analysing the confusion

Table 1. Training attacks grouped according to Weber-Kendal taxonomy

Group	**Attacks**
Probe	*Ipsweep, Nmap, Portsweep, Satan*
DoS	*Land, Neptune, Ping of Death, Smurf, Teardrop*
R2L	*Ftpwrite, Guess password, Phf, Spy*
U2R	*Perl*

Table 2. Performance results

	PC%	**FP%**	**FN%**	**NA%**	**cost**
KDDCup1999 winner	92.71	0.11	6.59	7.03	0.2331
K-means - *schema 1*	86.71	6.03	1.42	80.69	0.2487
K-means - *schema 2*	88.58	1.54	6.11	20.70	0.2635

Table 3. Cost Matrix

		Assigned				
		Normal	**U2R**	**R2L**	**Probing**	**DoS**
True	**Normal**	0	2	2	1	2
	U2R	3	0	2	2	2
	R2L	4	2	0	2	2
	Probing	1	2	2	0	2
	DoS	2	2	2	1	0

matrix, we also noted that a number of patterns belonging to the DoS class were labelled as Probe. Therefore, in order to overcome this problem, we set up an additional threshold (0.515) to be used in the decision process according to the values in the decision template. The attained new decision schema is shown in Figure 2. In the following, we will refer to this schema as *schema 1*. Afterwards, a further modification to the decision criteria has been applied. We labelled a pattern as R2L only if the maximum output of classifiers trained on R2L attacks was greater then a suitable threshold (0.515), and if the number N of classifiers which produced an output higher than n_{th}=0.5 was greater than 6. In the following we will refer to this decision schema as *schema 2* (see Figure 2).

Table 2 reports the performance results in terms of the percentage of correctly classified patterns **PC**, the false positive (false alarm) rate **FP**, the false negative (missed alarms) rate **FN**, the detection rate related to new attacks (i.e. test patterns related to attack types never seen during training) **NA**, and the average classification cost computed according to the cost matrix shown in Table 3. (The cost is computed as in [13]). Results are compared to the KDD Cup 1999 winner classifier (first row in table 2). It can be easily seen that our classifier ensemble performs better than the KDD Cup 1999 winner in terms of the false negatives rate and of the new attacks detection rate, especially in case of decision schema 1. Nevertheless, the proposed ensemble performs worse than the KDD Cup 1999 winner in terms of the percentage of false positives and of the overall cost, either

adopting decision schema 1 or 2. This reflects the inevitable trade-off between the false positive rate and the attack signature generalization ability.

6 Conclusions

In this paper, we have proposed a classifier ensemble architecture composed by one-class classifiers specialized in discriminating between patterns related to a specific attack class and patterns related to something else (i.e. patterns related either to normal usage or to different attack types). Combining the output of those one-class classifiers specialized in recognizing attacks belonging to a given attack class allowed us to attain an IDS based on *generalized attack signatures*. The proposed approach aimed at constructing a misuse-based Network Intrusion Detection System (NIDS) having a higher generalization ability than conventional signature-based NIDS. Experiments were carried out on the KDD Cup 1999 dataset. Performance results reflect the inevitable trade-off between the false positive rate and the *attack signature* generalization ability.

References

1. Axelsson S. *A preliminary attempt to apply detection and estimation theory to intrusion detection*. Technical report, Dept. of Computer Engineering, Chalmers Univerity of Technology, Sweden, March 2000.
2. McHugh J. *Intrusion and Intrusion Detection*. International Journal of Information Security, 2001, Vol. 1(1), 14-35.
3. Giacinto G., Roli F., Didaci L. *Fusion of multiple classifiers for intrusion detection in computer networks*. Pattern Recognition Letters, 2003, Vol. 24(12), 1795-1803.
4. Ryan J., Lin M.J., Miikkulainen R. *Intrusion Detection with Neural Networks*. In: Advances in Neural Information Processing Systems 10, 1998, M. Jordan et al., Eds., Cambridge, MA: MIT Press, 943-949.
5. Cordella, Limongiello, Sansone. *Network Intrusion Detection by a Multi-Stage Classification System*. Proceedings of the 5th International Workshop MCS 2004, 324-333.
6. D. Weber. *A Taxonomy of Computer Intrusions*. Master's thesis Massachussets Institute of Technology, 1998.
7. K. Kendall. *A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems*. Master s thesis, Massachussets Institute of Technology, 1999.
8. *KDD Cup 1999 dataset*. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
9. Northcutt S., Novak J., *Network Intrusion Detection (2 ed)*. New Riders Pub, 2001.
10. Lee W., Stolfo S.J., *A framework for constructing features and models for intrusion detection systems*. ACM Trans. on Inform. and System Security, 2000, 3(4), 227-261.
11. D. Tax. *One-class classification*. PhD thesis, Technische Universiteit Delft, 2001.
12. L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.
13. C. Elkan. *Results of the KDD 99 Classifier Learning*. ACM SIGKDD Explorations, 2000, Vol. 1(2), 63-64.

Combining Genetic-Based Misuse and Anomaly Detection for Reliably Detecting Intrusions in Computer Networks*

I. Finizio, C. Mazzariello, and C. Sansone

Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli "Federico II"
Via Claudio 21, I-80125 Napoli, Italy
{ifinizio, cmazzari, carlosan}@unina.it

Abstract. When addressing the problem of detecting malicious activities within network traffic, one of the main concerns is the reliability of the packet classification. Furthermore, a system able to detect the so-called *zero-day attacks* is desirable. Pattern recognition techniques have proven their generalization ability in detecting intrusions, and systems based on multiple classifiers can enforce the detection reliability by combining and correlating the results obtained by different classifiers.

In this paper we present a system exploiting genetic algorithms for deploying both a misuse-based and an anomaly-based classifier. Hence, by suitably combining the results obtained by means of such techniques, we aim at attaining a highly reliable classification system, still with a significant degree of new attack prediction ability. In order to improve classification reliability, we introduce the concept of rejection: instead of emitting an unreliable verdict, an ambiguous packet can be logged for further analysis. Tests of the proposed system on a standard database for benchmarking intrusion detection systems are also reported.

1 Introduction

The most common and best known tools used to ensure security of companies, campuses and, more in general, of any network, are Firewalls and Antiviruses. Though famous and well known, such tools alone are not enough to protect a system from malicious activities, and basing one's own site's security on the deployment of these instruments relies on the idea that intrusion prevention will suffice in efficiently assuring data availability, confidentiality and integrity. Indeed, an interesting idea about intrusions is that they will sooner or later happen, despite the security policy a network administrator deploys. Based on such assumption, many researchers started to develop systems able to successfully detect intrusions and, in some cases, trace the path leading to the attack source.

* This work has been partially supported by the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) in the framework of the FIRB Project "Middleware for advanced services over large-scale, wired-wireless distributed systems (WEB-MINDS)".

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 66 – 74, 2005.

On the basis of the information sources analyzed to detect an intrusive activity, the Intrusion Detection Systems (IDS) can be grouped into different categories. In the following, we will concentrate our attention on Network-based IDS (N-IDS) [1]. N-IDS analyze packets captured directly from the network: by setting network cards in promiscuous mode, an IDS can monitor traffic in order to protect all of the hosts connected to a specified network segment.

Depending on the detection technique employed, they can be roughly classified [2] as belonging to two main groups as well. The first one, that exploits signatures of known attacks for detecting when an attack occurs, is known as *misuse* (or *signature*) *detection* based. IDS's that fall in this category are based on a model of all the possible misuses of the network resources. The completeness request is actually their major limit [3].

A dual approach tries to characterize the normal usage of the resources under monitoring. An intrusion is then suspected when a significant difference from the resource's normal usage is revealed. IDS's following this approach, known as *anomaly detection* based, seem to be more promising because of their potential ability to detect unknown intrusions (the so-called *zero-day* attacks). However, there is also a major challenge, because of the need to acquire a model of the normal use general enough to allow authorized users to work without raising false alarms, but specific enough to recognize unauthorized usages [4,5].

The network intrusion detection problem can be also formulated as a binary classification problem: given information about network connections between pairs of hosts, the task is to assign each connection to one out of two classes that represent normal traffic conditions or an attack. Here the term "connection" refers to a sequence of data packets related to a particular service, such as a file transfer via the ftp protocol. In this framework, several proposals have been made in order to extract high-level features from data packets [6,7]. Each network connection can be then described by a "pattern" to be classified and a pattern recognition approach can be followed. The main advantage of such an approach is the ability to generalize exhibited by pattern recognition systems. They are able to detect some novel attacks, since different variants of the same attack will be typically described by very similar patterns. Moreover, the high-level features extracted from connections relative to a totally new attack should exhibit a behavior quite different from those extracted from normal connections. Summarizing, there isn't the need of a complete description of all the possible attack signatures. This overcomes one of the main drawbacks of the misuse detection approach.

Different pattern recognition systems have been reported in the recent past for realizing an IDS, mainly based on neural network architectures [4,8,9]. In order to improve the detection performance, approaches based on multi-expert architectures have been also proposed [10,11,12].

However, one of the main drawbacks occurring when using pattern recognition techniques in real environments is the high false alarm rate they often produce [10]. This is a very critical point in a real environment, as pointed out in [13].

In order to realize an IDS that is capable of detecting intrusion by keeping the number of false alarms as low as possible, in this paper we propose a genetic-based

system that tries to combine some of the peculiarities of the misuse and of the anomaly detection approaches.

In particular, starting from the features proposed in [6], a genetic algorithm is used to generate two distinct sets of rules. The first set is devoted to individuate normal traffic connections (as in an anomaly detection approach), while the second one is suited for detecting attacks (following the misuse detection paradigm). A connection is then classified as an attack if it matches almost one of the rules of the second set and no one of the first set. On the other hand, a connection is labeled as normal if it matches almost one of the rules devoted to detecting normal traffic and no one of those generated for characterizing attacks. In all the other cases, the connection is rejected by the system, since it cannot be correctly classified with an high reliability. This permits to reduce the number of false alarms. Note that reject, in this case, means that the data about a 'rejected' connection are only logged for further processing, without raising an alert for the system manager.

It is worth noticing that other rule-based classifiers have been employed in an IDS (for example RIPPER [14], used by Lee and Stolfo in [6]). They, however, follow only the misuse detection approach, thus giving rise to a false alarm rate that cannot be acceptable in a typical real environment.

The organization of the paper is as follows: in Section 2 the proposed system is presented, while in Section 3 tests on a standard database used for benchmarking IDS are reported, together with a comparison of the proposed system with other pattern recognition systems. Finally, some conclusions are drawn.

2 A Genetic Approach for Generating Rules

As stated in the introduction, the proposed system is a rule-based one. Two sets of rules are generated, each one devoted to individuate a specific class, namely attacks or normal traffic. In order to classify a new traffic connection, the results of the two rule-based classifiers are suitably combined by means of a decision engine. In particular, if the feature vector describing a connection matches one of the rules related to the normal traffic and does not match any of the rules related to the attack class, it is attributed to the normal traffic class. Vice versa, if it matches almost a rule describing attacks and no one of the rules describing normal traffic, an alert is raised. In all the other cases, data about the connections are just logged for further processing (see Fig. 1).

Each set of rules is generated by means of a genetic algorithm based on a particular structure of the chromosomes. Such a structure was developed for suitably representing the boundaries of the region of the *n*-dimensional space containing the feature vectors representing the network connection belonging to the class the chromosome refers to.

Each chromosome consists of *n* genes. Each gene is associated to an element of the feature vector to be classified and is composed by a pair of values, x_{iMIN} and $x_{iMAX.}$ Such values represent, respectively, the lower and the upper limit of an interval. If the values of all the elements of the feature vector fall within the limits specified by the corresponding genes of a chromosome, this feature vector is attributed to the class the

chromosome refers to. The minimum value that x_{iMIN} can assume is $-\infty$, while the maximum value that x_{iMAX} can assume is $+\infty$. The conversion from a rule to a chromosome and vice versa is immediate since they are simply two different ways to represent decision regions (see Fig. 2).

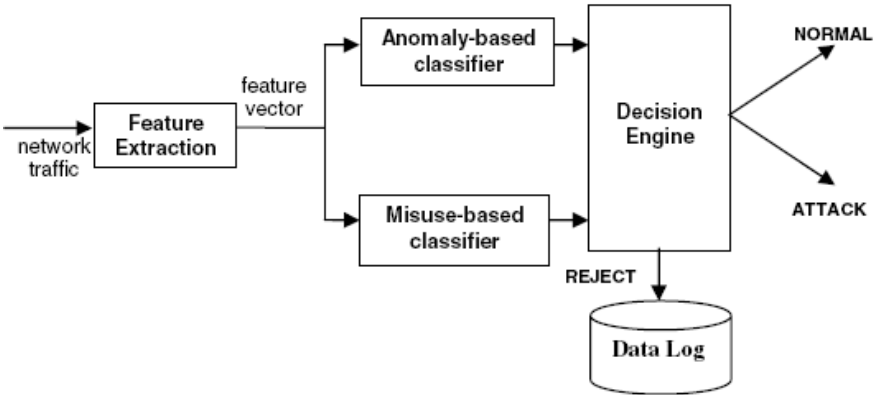


Fig. 1. A sketch of the proposed architecture for intrusion detection

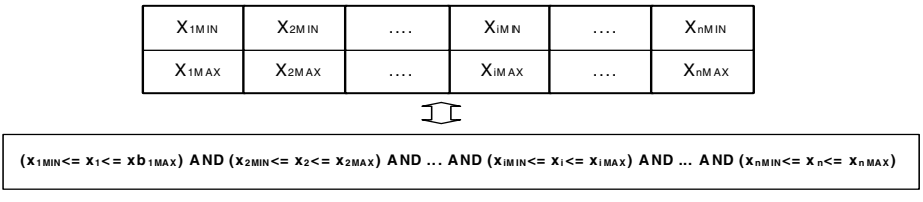


Fig. 2. Structure of a chromosome and the associated rule

Thus, each chromosome represents a hyper-region in the *n*-dimensional space. The aim of the proposed algorithm is to identify the region which the feature vectors belonging to a given class (normal or attack) lie into, that is, to select the corresponding chromosomes. The first step consists in the generation of an initial population of chromosomes, by assigning to each gene pairs of pseudo-random values. The assigned values are randomly selected from the set of all the values assumed within the whole training set by the corresponding elements, with the addition of the $-\infty$ and $+\infty$ values. The constraint to be observed is that, for each gene, the value of x_{iMIN} cannot be greater than the value of x_{iMAX}. After the computation of the fitness value for each generated chromosome, the reproduction process starts. A hybrid method was designed for the selection of the parents. This method can be considered as a binary selection double tournament with steady-state replacement. The algorithm randomly selects two pairs of chromosomes and, within each pair, compares the fitness. The fittest chromosomes of each selected pair are selected for reproduction and their two children take the place of the loser chromosomes. This

technique promotes the elitism. In fact, the best chromosomes in each binary tournament are always winning and never substituted. After each step of the reproduction only the new individuals' fitness is recomputed and they are immediately ready for the reproduction. By using such a mating strategy, it is possible to use a promising individual just as soon as it is created. The used fitness function is:

$$Fitness = \frac{k_1 \cdot (neg + 1 + \left(\frac{elem}{k_3} \right))}{k_2 \cdot pos + 1} \tag{1}$$

where k_1, k_2 and k_3 are three parameters whose optimal values were fixed by an experimental investigation, *pos* and *neg* are respectively the number of feature vectors belonging to the training set which are correctly classified by the rule associated to the chromosome and the number of feature vectors belonging to the same set which are misclassified, *elem* is the number of elements of a feature vector not generalized to *any*. We assume that an element is generalized to *any* when the corresponding gene covers the whole set of real numbers, that is, when $x_{iMIN}=-\infty$ and $x_{iMAX=}+\infty$. Lower fitness values correspond to better chromosomes; therefore, in the comparison between two chromosomes, the one with the lower fitness is chosen. We insert the number of elements whose corresponding gene is not generalized to *any* in the fitness function in order to favour less complicated rules. Having fixed the value of all the other parameters, in fact, a rule with a simpler structure is associated to a chromosome characterized by a smaller value of the *elem* parameter. To obtain a behaviour as independent as possible from the training set used, we have adopted a uniform crossover strategy. For each reproduction step, a mask made by a pair of values for each gene of the chromosome is randomly generated. Each element of the mask contains the value 1 if it should prevail the present value in the first chromosome involved in the reproductive process, and 0 in the opposite case. The first one of two crossovers is carried out on the basis of such a mask. The second crossover is carried out using a mask obtained by complementing the previous one. It must be guaranteed that $x_{iMIN} <= x_{iMAX}$ also in the crossover phase.

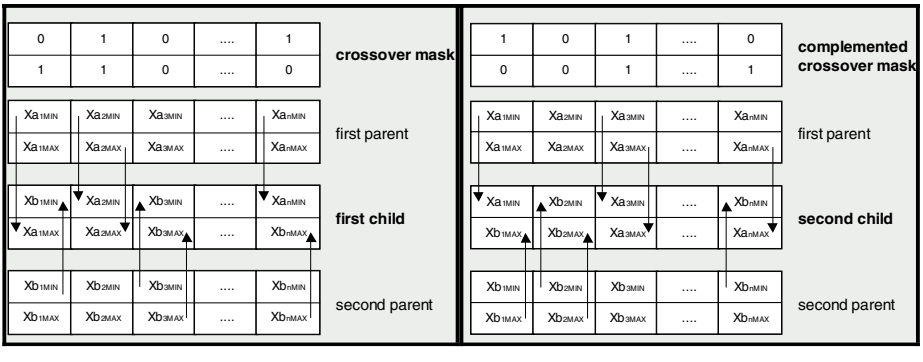


Fig. 3. The crossover mechanism

At the end of the crossover, the new chromosomes undergo the mutation process. A mutation mechanism with incremental probability was proposed, in order to fully exploit the peculiarity of both crossover and mutation. We initialize the mutation probability to a very low value, then we progressively increase it during the genetic analysis. This technique offers the advantage of fully exploiting the mutation capacity of moving a solution far from local maxima. Then, it avoids the problems of a slow convergence and, even though in a smaller measure, of a premature convergence. Once the convergence has been reached, the obtained chromosomes are chosen and translated into a rule. If the rule associated to such chromosomes is able to correctly classify all the training set data, the process ends, otherwise it restarts and tries to identify a new chromosome able to classify the feature vectors not covered by the previously selected chromosome. At the end of the process, the set of rules that describes the whole decision region is composed by all the rules corresponding to the selected chromosomes.

3 Experimental Results

The proposed system has been tested on a subset of the database created by DARPA in the framework of the 1998 *Intrusion Detection Evaluation Program*. It is made up of a large number of network connections related to normal and malicious traffic. This database was pre-processed at the Columbia University giving rise to a feature vector of 41 elements for each connection, according to the set of features defined in [6] and tailored for the intrusion detection problem. Each connection in the database is labelled as belonging to normal traffic or to an attack. It is worth noticing that the attack class is made up of different variants, each one exploiting different vulnerabilities of a computer network.

The 1999 KDD intrusion detection contest used a version of this dataset. Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it was a true Air Force environment, but peppered it with multiple attacks. Even if this database has been collected in 1998, and some criticisms have been expressed on it [15], it is still widely used for testing the performance of an IDS [9,16].

The results obtained by means of the proposed system are reported in terms of *i*) the overall error rate on the classified connections, *ii*) the false alarm rate and the missed detection rate on the classified connections, and *iii*) the reject rate.

In the following we present the results obtained by our classification method applied to two different network services (*smtp* and *http*) among those present in the DARPA database. Other services have also been experimented, but the obtained results are not reported here for the sake of brevity. The choice of designing a different classifier system for each service follows the so-called modular approach presented in [11], where the authors experimentally demonstrate the advantage, in terms of recognition performance, of an IDS that develops a different classification module for each one of the network services to be protected.

For each service, a separate feature selection process was performed in order to reduce the data dimensionality. In particular, we have adopted a Sequential Forward Selection strategy, with the Minimum Estimated Probability classification criterion.

Moreover, different values of the parameters k_1, k_2 and k_3 (see eq.1) have been tested. The selected value has been chosen so as to maximize the obtainable results on the training set.

3.1 *Smtp* Service

In this case the training data was made up of 9723 patterns related to different attack variants and to the normal class. The Test Set (TS) for this service is made up of 3261 patterns with 3207 normal connections and 54 attack packets. After the feature selection process, each connection was described by a 6-dimensional feature vector. In particular, *duration*, *flag*, *src_bytes*, *hot*, *count* and *dst_src_host_same_src* features were selected (see [6] for their meanings).

Table 1 shows the results achieved by the proposed system on the TS. These results have been obtained by averaging ten different trials of the genetic algorithm for generating the two sets of rules.

As it is evident, we reach an ideal performance in terms of both missed detections and false alarms on the connections classified by the system. This excellent result is paid with about a 9% of reject rate.

Table 1. Results obtained by the proposed system on the TS for the *smtp* service

Overall error	*False alarm rates*	*Missed Detection rate*	*Reject rate*
0.00 %	0.00 %	0.00 %	9.12 %

Moreover, it is interesting to note that among the classified connections the proposed system is able to correctly detect over the 96% of the attacks to the *smtp* service.

3.2 *Http* Service

The training data for this service in the DARPA database are made up of 64292 patterns. However, in [9] it has been demonstrated that a dataset of about 15% of the whole *http* data is sufficient for training classifiers. Therefore, only 8866 samples have been considered as training data. The test set for this service is made up of 40442 patterns with 1195 attack packets and 39247 normal connections. After the feature selection process each connection was described by a 6-dimensional feature vector. The selected features were the same of the *smtp* service.

Table 2. Results obtained by the proposed system on the TS for the *http* service

Overall error	*False alarm rates*	*Missed Detection rate*	*Reject rate*
0.08 %	0.08 %	0.06 %	9.67 %

Table 2 shows the results achieved by the proposed system on the TS. Also in this case, the results reported here have been averaged on ten different trials of the genetic algorithm for generating the two sets of rules. In this case the system exhibits a quite negligible percentage of false alarms and missed detections. On the other hand, it

must be noted that among the classified traffic connections, about the 44% of attacks were detected.

In order to make a comparison with other systems, it can be noted that the multi-stage classification system proposed in [12] achieved on the *http* connections a slightly higher false alarm rate (0.09%), while the multi-expert system proposed in [11] exhibited a false alarm plus missed detection rate of 0.54%. This confirms that our system is able to keep the number of false alarms low.

4 Conclusions

In this paper we proposed a genetic-based system for intrusion detection. A genetic algorithm is used for building two rule-based classifiers, a misuse-based one and an anomaly-based one. By suitably combining their *opinions* about each analyzed network connection, a decision engine improved the ability of the system in avoiding detection errors.

The proposed system showed a very encouraging behavior from the detection capability point of view. In particular, in case of the *smtp* service, we observe an error rate which is equal to 0%. On the other hand, we have a not negligible number of rejected packets.

Therefore, as a future development of the proposed architecture, we will work on the analysis of the rejected packets with slower but more accurate algorithms, in order to further improve the detection capability of the system.

References

1. G. Vigna, R. Kemmerer, “Netstat: a network based intrusion detection system”, Journal of Computer Security, vol. 7, no. 1, 1999.
2. S. Axelsson, Research in Intrusion Detection Systems: A Survey, TR 98-17, Chalmers University of Technology, 1999.
3. R. Kumar, E.H. Spafford, “A Software Architecture to Support Misuse Intrusion Detection”, in Proceedings of the 18th National Information Security Conference, pp. 194-204, 1995.
4. A.K. Ghosh, A. Schwartzbard, “A Study in Using Neural Networks for Anomaly and Misuse Detection”, Proc. 8'th USENIX Security Symposium, Aug. 26-29 1999, Washington DC.
5. T. Lane, C.E. Brodley, “Temporal Sequence learning and data reduction for anomaly detection”, ACM Trans. on Inform. and System Security, vol. 2, no. 3, pp. 295-261, 1999.
6. W. Lee, S.J. Stolfo, “A framework for constructing features and models for intrusion detection systems”, ACM Transactions on Inform. System Security, vol. 3, no. 4, pp. 227-261, 2000.
7. M. Esposito, C. Mazzariello, F. Oliviero, S.P. Romano, C. Sansone, “Real Time Detection of Novel Attacks by Means of Data Mining Techniques”, Proceedings of the 7th International Conference on Enterprise Information Systems, Miami (USA), May 24-28, 2005 (in press).

8. S. C. Lee, D.V. Heinbuch, "Training a neural Network based intrusion detector to recognize novel attack", IEEE Trans. Syst, Man., and Cybernetic, Part-A, vol. 31, pp. 294-299, 2001.
9. M. Fugate, J.R. Gattiker, "Computer Intrusion Detection with Classification and Anomaly Detection, using SVMs", International Journal of Pattern Recognition and artificial Intelligence, vol. 17, no. 3, pp. 441-458, 2003.
10. G. Giacinto, F. Roli, L. Didaci, "Fusion of multiple classifiers for intrusion detection in computer networks", Pattern Recognition Letters, vol. 24, pp. 1795-1803, 2003.
11. G. Giacinto, F. Roli, L. Didaci, "A Modular Multiple Classifier System for the Detection of Intrusions", Lecture Notes in Computer Science vol. 2709, pp. 346-355, 2003.
12. L. P. Cordella, A. Limongiello, C. Sansone, "Network Intrusion Detection by a Multi Stage Classification System", Lecture Notes in Computer Science vol. 3077, Springer, Berlin, pp. 324-333, 2004.
13. S. Axelsson, "The Base-Rate Fallacy and the Difficulty of Intrusion Detection", ACM Trans. on Information and System Security, vol. 3, no.3, pp. 186-205, 2000.
14. W.W. Cohen, "Fast effective rule induction". In Proc. of the 12th International Machine Learning Conference, Morgan Kaufmann, 1995.
15. J. McHugh, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory", ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262-294, 2000.
16. Y. Liu, K. Chen, X. Liao, W. Zhang, "A genetic clustering method for intrusion detection", Pattern Recognition vol. 37, 2004.

EnFilter: A Password Enforcement and Filter Tool Based on Pattern Recognition Techniques

Giancarlo Ruffo and Francesco Bergadano

Dipartimento di Informatica,
Università degli Studi di Torino
{ruffo, bergadano}@di.unito.it

Abstract. EnFilter is a Proactive Password Checking System, designed to avoid password guessing attacks. It is made of a set of configurable filters, each one based on a specific pattern recognition measure that can be tuned by the system administrator depending on the adopted password policy. Filters use decision trees, lexical analysers, as well as Levenshtein distance based techniques. EnFilter is implemented for Windows 2000/2003/XP.

1 Introduction

Passwords are considered weak when they are (1) guessable (i.e., phone numbers, boyfriend names, car's numberplate, ...), (2) not resistent against dictionary driven attacks (see Crack software [6]), or (3) with low entropy [10], and, as a consequence, not secure in terms of brute force attacks.

Proactive password checking is considered the best approach [3,7,9] for avoiding selection of weak passwords. Quite surprisingly, the literature on this field is limited, if one considers the great practical importance of this problem. This does not imply that acceptable solutions to the given problem do not exist [1,2,3,7,9,10]. In Particular, the EnFilter system, which is presented in this paper, is partially based on ProCheck [1], that provides a solution for Unix systems. Some key features have been changed, as explained in Section 4, and the system has been engineered for Windows 2000/2003/XP.

2 Password Filters and Proactive Checking

Spafford suggests that proactive checking can take advantage of Bloom filters [9]; Davies e Ganesan adopt a markovian model in Bapasswd [3], and Nagle proposes a simple, but effective test based on a lexical analyser in [7]. With ProCheck, proactive checking is reduced to a Pattern Recognition problem, where the task is to learn the rules to classify passwords as good or bad. These rules can be represented by means of decision trees, built by classical induction algorithms ID3-like; in fact, ProCheck uses C4.5 of Quinlan [8] to build a decision tree from a "crack" dictionary (a list of examples of bad passwords), and from a randomly generated file of "good" passwords.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 75–82, 2005.

In [1], these approaches are compared on the basis of the compression rate of the given dictionary and the time taken by the classifiers to decide if a password is good or bad. Of course, another important parameter is given by the classification error percentage (the sum of the rate of false negatives and false positives).

as reported in the following sections, Enfilter uses a "Dictionary Filter" based on decision tree classification. The implementation of such a filter is a scalable generalization of ProCheck.

Even though ProCheck is still referenced in [10] as the most efficient solution w.r.t. classification times and space required for storing the compressed dictionary, in [2] learning is phase is refined in order to further improve the compression rate of the derived decision tree. Yan in [10], addresses one of the most important flaws of existing proactive checkers (including ProCheck): they fail to catch weak passwords like `ca12612, hhiijjkk, a1988b53, 12a3b4c5` and `12a34b5`. In fact, such passwords can be considered invulnerable against a dictionary attack, but their *low-entropy* makes them vulnerable to brute force attacks.

As a further consideration, another weakness of ProCheck is given by its poor scalability: even if strong against attacks driven with the same dictionary used during the learning phase, ProCheck fails with passwords that trivially were not given to the inductive learner (as, for example, passwords contained in a dictionary of another language).

ProCheck has been shown to be noisy resistant in [1], i.e., classification must be strong against passwords that are slightly different from those given to the learner. But this is not enough: what if an administrator requires that the system under her responsibility is strong against a particular dictionary? How to improve the proactive checking with additional filters based on some lexical password features (for example, in a given environment we can ask the user to adopt passwords containing at least one upper case letter, two special characters and one digit)? In other words, the proactive checker should combine an efficient and complete defence against dictionary attack and a configurable filter that rejects low-entropy passwords.

Finally, another weakness of ProCheck has been highlighted during its usage at our department, even if it does not concern with a security flaw. ProCheck has been running at the student's laboratory (about 500 users) since four year, but users sometimes complain because too many *difficult* passwords were not accepted. When the checker is too severe, productivity of the user is reduced, and money is lost. The reason behind this excessive severity is explained in Section 5.1, where EnFilter's solution is proposed.

In the Microsoft Windows framework, some commercial tools allow the insertion of additional filters based on lexical rules, but to the best of our knowledge, no current access control package is able to proactively check against both a set of dictionaries and configurable filters.

EnFilter is a tool that allows for the extension of the proactive checking features by means of filters with different characteristics. This control is com-

pletely integrated in the Windows framework, with a proper management of the Security Account Manager (SAM) database and by means of the Notification Packages [5].

3 Dictionary Filter Based on Decision Tree Classification

The Dictionary Filter described in Section 4, is based on a decision tree classifier. This proactive checker filters out passwords that are classified as "not resistent" against a dictionary attack.

We view the training phase of a password checker as a particular Pattern Recognition problem. More precisely, we would like to use the dictionaries as sources of *positive examples* (i.e., belonging to the dictionary), and learn more concise descriptions that classify passwords as either positive or negative. We also choose to create an explicit set of *negative examples* (i.e., not belonging to the dictionary) by generating random passwords that do not belong to the dictionaries. Examples are given all at once and no new dictionaries can be added without repeating the training phase. This fits into the standard framework of one-step learning from positive and negative examples.

We chose a *decision tree* representation due to three reasons: (1) word membership is a simple problem and does not need more expressive formalisms, (2) excellent systems have been developed and implementations are available and (3) decision trees are likely to achieve greater compression on word membership problems, because prefixes common to many words need be stored only once.

Words will be described by means of so-called *attributes*, i.e. functions that take a finite number of *values*. A node of a decision tree will correspond to some attribute, and the arcs from this node to its children correspond to particular values of the attribute. Leaves will be associated to a classification, positive or negative. A decision tree can then be used for classifying a word w as follows: we start from the root of the tree, and evaluate the corresponding attribute for w obtaining value v; then we follow the arc labelled by v and reach the corresponding node; then, we repeat the procedure until a leaf is reached, and output the associated classification.

As an example, suppose we have a dictionary containing just two bad passwords: "ab" and "cow". Suppose also that we generate two random passwords: "w31" and "exw". Then, we label "ab" and "cow" as positive examples, and "w31" and "exw" as negative examples. In practice, dictionaries with million of words, each with 6 characters or more, are largely used to train the system (e.g, as described in [1] ProCheck was trained on a dictionary of 3,215,846 *bad* words plus a list with the same number of examples containing random *good* words). Now, suppose we describe these examples with 3 attributes:

a1 - equals 0 if the first character is a vowel, 1 otherwise;
a2 - equals 0 if the second character is a vowel, 1 otherwise;
a3 - the length of the word.

Ley $D1$ be a decision tree that will first consider the length of the word (i.e., the value of $a3$ is checked). If the length is less than 3, the example is classified

positive, otherwise $D1$ will examine the second character, to see whether it is a vowel. If $a2 = 0$, the example is classified as positive, otherwise it is negative, e.g, $D1$ classifies examples"exw" and "w31" as negative. $D1$ is an acceptable solution of this simple classification problem.

A reference system for learning decision trees is the very well known ID3 [8]. Its basic top-down learning strategy is followed in most other methods. Initially, the tree is empty and an attribute need be selected for the root node. All positive and negative examples are associated to the root node of the tree. Among the possible attributes, ID3 chooses one that maximizes an information-theoretic quantity called the *gain*.

The gain of an attribute a is computed as follows. Suppose the father node is associated to p positive and n negative examples, with $p + n = t$. Define the *information* represented by this partitioning of positive and negative examples as

$$I(p, n) = -\frac{p}{t} log_2 \frac{p}{t} - \frac{n}{t} log_2 \frac{n}{t} \tag{1}$$

The information after attribute a has been selected is then computed as the weighted sum of the information corresponding to the children nodes:

$$I(a) = \sum_{i=1}^{s} \frac{t_i}{t} I(p_i, n_i) \tag{2}$$

where there are s possible values of a, and p_i (n_i) out of the p positive examples have the i-th value for attribute a. Again, $t_i = p_i + n_i$. The *gain* of attribute a is then defined as $I(p, n) - I(a)$.

An important topic in decision tree learning, that is very relevant to the present study, goes under the name of *pruning*. Decision trees that are learned with the above technique will correctly classify all given examples, if the attributes are sufficient to discriminate one class from the other. In other words, the *observed error rate* will be 0. However, if the tree is large, it may happen that some leaves, or even large subtrees, are only useful for a limited set of examples.

In the given domain, predictive power of pruned decision tree has been showed to perform well in case of noisy password in [1], for instance, only 0.99% of the words of a file with 150,000 noisy words were classified as good passwords. Other experiments lead also to very encouraging results. Moreover, if pessimistic pruning is adopted at its best, compression rate can be further reduced [2].

Previous experiments performed on ProCheck showed that this approach behaves well in terms of classification time (i.e., linear w.r.t. the password length), compression and error rate: a decision tree classifier of size 24, 161 bytes was obtained from a dictionary of "bad" passwords of about 28 MB, with an error-rate of 0.5%. Using an exception file of 171 KB containing the words in the dictionary that are incorrectly classified as good passwords by the decision tree classifier, ProCheck reduces the error to one-sided and is equal to 0.32 (i.e., misclassified randomly generated password not belonging to the dictionary, but given to the learner).

4 EnFilter: How It Works

EnFilter maintains security and reliability properties of ProCheck, but coming through the deficiencies located above. EnFilter[1] is: (1) Highly scalable; (2) Designed for Microsoft Windows 2K/XP/2003; (3) Manageable by a simple GUI.

Once EnFilter has been installed, the user can select one of the following options:

1. Activating or deactivating EnFilter controls that are executed when a password change request is sent to the Notification Package (i.e., a user is attempting to change his/her own password).
2. Testing the strength of a password accordingly the current activated filters.
3. Setting up the installed filters, including those based on different language dictionaries.

The user interacts with an application that can be run from the Windows control panel. This application is a GUI implemented with Microsoft Visual J++. EnFilter calls a procedure contained in a DLL that implements a decision tree classifier. This is a classification procedure, which reads the stored decision trees, each for a given dictionary. Enfilter.dll is also responsible of checking if the password under test can be considered strong against all the activated filters. It implements (and exports) the PasswordChangeNotify, PasswordFilter e InitializeChangeNotify functions, as requested to the developers [5]. As a consequence, EnFilter is activated also when an user asks for a password change.

At the present time, together with the dictionary filter, that allows the user to check the passwords against a set of different language dictionaries, EnFilter supports a configuration filter checking for at least n (with $n > 0$) alphabetical letters, a filter checking for at least n (with $n > 0$) digits, and a filter checking for at least n (with $n > 0$) special characters. EnFilter can also check for a minimum length of the password. In this way, the system administrator is able to set up different access control policies against attacks based on password guessing. In Table 1[2], we show some examples of passwords that can be accepted or rejected by EnFilter depending on the different filters that can be activated.

5 Filters Description

The system administrator is responsible of enabling the given filters and, when needed, of configuring them. For brevity, we do not discuss other simpler (but effective) filters, e.g. ones based on lexicographical rules.

[1] You can find further detail and the instruction to download EnFilter at *http://security.di.unito.it*.

[2] Values in the table must be interpreted as it follows: "NO" - the password is filtered out; "OK" - the password passes the test defined by the given filter. As a consequence, it is considered strong.

Table 1. En example of different password tests of the user "scrooge"

Password	Windows Standard Check	Applied Filters			
		Italian dictionary	English dictionary	At least 1 special character	At least 2 digits
Scrooge	NO	NO	NO	NO	NO
2Skulls	OK	OK	NO	NO	NO
xltrk9u	OK	OK	OK	NO	NO
xltr+k9u	OK	OK	OK	OK	NO
Xty89'ad76	OK	OK	OK	OK	OK
S(r0oge1	OK	OK	OK	OK	OK

5.1 Dictionary Filter

This filter is strongly based on the decision tree approach described in Section 3, with the following important differences w.r.t. ProCheck:

1. a new set of attributes was used. Learned decision tree is able to classify words using an attribute a_i (where $i = 1, ..., 12$) for each letter i in the word. Each attribute can have 27 different values, given by the corresponding letter (e.g., $a_i = 1$, if the i-th character is 'a' or 'A', $a_i = 2$, if the i-th character is 'b' or 'B', ..., else $a_i = 0$). For example, password $Scr0ge1$ is represented with the following attribute-value vector: $< 19, 3, 18, 0, 7, 5, 0, 0, 0, 0, 0, 0 >$.
2. We included a different decision tree for each dictionary. In the current distribution we *compressed* four natural language dictionaries: Italian, English, Spanish and German.

The second feature comes from the scalability and configurability requirements that can be addressed from a typical system administrator, and therefore it is just a direct consequence of making EnFilter practical and usable in the real world. The first difference is more evident if we recall (one of) the attribute-value representation that has been adopted in ProCheck:

a_n = value of letter number n in the word *(for n between 1 and 12)*, where values are as follows: 1 for vowels; 2 for *n, m, r, l, c, g, k, x, j, q,* and *h*; 3 for *t, d, b, p, f, v, w, s,* and *z*; 4 for digits; 5 for all other characters
a_{13} = 0 if the word contains a special character, 1 otherwise;
a_{14} = number of capital letters in the word.

This set of attributes results in a big compression (i.e., characters are grouped together, and therefore the number of arcs in the final decision trees is dramatically reduced), but it has two heavy drawbacks:

1. The learning phase is strongly biased on attributes a_{13} and a_{14}: randomly generated words contain much more upper case letters and special characters than strings listed in a crack dictionary. The learned decision tree is too restrictive, because we wish to let words such as *xltrk9u* pass the dictionary filter and leave to the administrator the choice of limiting the minimum number of upper case and special characters using the lexicographical filters introduced early in this section.

2. Too many words collapse in the same description, creating misleading collisions. For example, the word *Teorema* has the same ProCheck-representation (i.e., $< 3, 1, 1, 2, 1, 2, 1, ?, ?, ?, ?, ?, 1, 1 >$[3]) of the string *woIxaku*, that it does not belong to any of the used dictionaries.

With the new set of attributes, EnFilter performed well in terms of compression and error rate. In fact, we observed an average of 0.5% of false positives[4], with a confidence interval of $[0.35, 3.84]$. These results are comparable with ProCheck, with its average of 0.53% of false positives, and with a confidence interval of $[0.25, 0.67]$. Moreover, we have an identical false negatives rate in both systems. Despite to these encouraging results in terms of error percentage, the compression, as expected, decreased from a rate of 1000 to 1, to a ratio of 100 to 1.

It goes without saying that the compression factor is not so important as some years ago, because of the bigger capacity and lower cost of current memorization devices. Nevertheless, the EnFilter distribution is very small in size. It includes the following decision trees: English.tree (394 KB), Italian.tree (241 KB), Spanish.tree (55 KB), German.tree (69 KB), that can be further compressed in a zipped archive. Finally, the entire installation package is sized 621 KB.

5.2 Directory Service Filter

Another filter currently under testing is based on another information theoretic quantity called *Levenshtein distance* [4]. It returns the distance between two strings, which is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a character.

Let s and t be two strings of length n and m, respectively. Given two characters s_i and t_j, which are respectively the i-th and the j-th character of s and t, let us define function r as follows: $r(s_i, t_j) = 0$, if $s_i = t_j$; $r(s_i, t_j) = 1$, otherwise. Now, we build a matrix D of integers, with dimension $(n+1)(m+1)$. Values in T are defined recursively as it follows:

$$
\begin{aligned}
D(i,0) &= i,\ i = 0, 1, ..., n \\
D(0,j) &= j,\ j = 0, 1, ..., m \\
D(i,j) &= min(D(i-1,j)+1, D(i,j-1)+1, D(i-1,j-1)+r(s_i,t_j)),\ i \neq 0 \wedge j \neq 0
\end{aligned}
$$

The element $D(n+1, m+1)$ gives the Levenshtein distance $d(s,t)$ between strings s and t. For example:
$d(\text{"}scrooge\text{"}, \text{"}scrooge\text{"}) = 0$, $d(\text{"}scrooge\text{"}, \text{"}S(r0oge1\text{"}) = 5$, and so on.

We used this distance in order to validate the password against the many values retrievable from the user's entry of a given Directory Service (e.g., LDAP,

[3] The value ? is used when the corresponding attribute is undefined. In this example, the word "Teorema" is 7 character long, and therefore attributes $a_8, ..., a_{12}$ are undefined.

[4] false negatives, i.e., words in the dictionary erroneously classified as good passwords, are not considered, because we use an exception file for them.

Microsoft Directory Service). The filter accepts only the passwords having a Levenshtein distance greater than a given threshold ε from all the string values contained in the user's entry of the local directory server. Such information usually contains many user's personal information, like birthdate, phone numbers, address, and so on. Observe that the degree of variation can be properly configured, i.e., threshold ε can be set by the system administrator, even if a default value of 3 is given if this filter is activated.

6 Conclusion

EnFilter, a proactive password checker designed and implemented for Microsoft Windows platforms, has been introduced. It is a configurable and scalable tool, which leaves the administrator the responsibility of adequating filters to the password policies. In particular, the "Dictionary Filter" improves the previous results obtained with ProCheck, reducing false alarms caused by the absence of special characters or upper cases in the checked password. Moreover, EnFilter does not increase the false negative rate, that is anyhow reduced to zero by adopting small sized exception files.

Acknowledgements

The authors wish to thank the anonymous referees for helping them to significantly improve the quality of the presentation. This work has been partially financially supported by the Italian FIRB 2001 project number RBNE01WEJT "Web MiNDS".

References

1. Bergadano, F., Crispo, B., and Ruffo, G. High Dictionary Compression for proactive password checking on ACM TISSEC, 1(1), Nov. 1998.
2. Blundo C., D'Arco P., De Santis A., Galdi C., Hyppocrates: a new proactive password checker, The Journal of Systems and Software, N. 71, 2004.
3. Davies, C. and Ganesan, R. Bapasswd: a new proactive password checker In Proc. of 16th NIST-NCSC National Computer Security Conference (1993).
4. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. Sov. Phys. Dokl., 6:707-710, 1966.
5. Microsoft Knowledge Base HOWTO: Password Change Filtering & Notification in Windows NT - article n. 151082
6. Muffett, A. Crack 4.0, 5.0 11.
7. Nagle, J. B., An obvious password detector. In USENET news - comp.sources.unix 16 (60), 1988.
8. Quinlan, J. R. C4.5: Programs for Machine Learning Morgan Kaufmann, San Mateo, CA.
9. Spafford, E.H. OPUS: Preventing Weak Password Choices Computers and Security, 11, (1992) pp.273-278.
10. Yan, J. A Note on Proactive Password Checking, ACM New Security Paradigms Workshop, New Mexico, USA, September 2001.

Analyzing TCP Traffic Patterns Using Self Organizing Maps

Stefano Zanero

D.E.I.-Politecnico di Milano,
via Ponzio 34/5 - 20133 Milano Italy
zanero@elet.polimi.it

Abstract. The continuous evolution of the attacks against computer networks has given renewed strength to research on anomaly based Intrusion Detection Systems, capable of automatically detecting anomalous deviations in the behavior of a computer system. While data mining and learning techniques have been successfully applied in host-based intrusion detection, network-based applications are more difficult, for a variety of reasons, the first being the curse of dimensionality. We have proposed a novel architecture which implements a network-based anomaly detection system using unsupervised learning algorithms. In this paper we describe how the pattern recognition features of a Self Organizing Map algorithm can be used for Intrusion Detection purposes on the payload of TCP network packets.

1 Introduction

The continuous evolution of the attacks against computer networks and systems has given renewed strength to research on "anomaly based" Intrusion Detection Systems (IDS). The "misuse based" approach, which tries to define what constitutes an attack in order to detect them, has been widely successful in the past, but is increasingly showing its limits.

The growing number of attacks requires a continuous update of the knowledge base of misuse based IDSs. In addition, there is also an unknown number of discovered but undisclosed vulnerabilities (the so called "zero-days") that are not available for analysis and inclusion in the knowledge base. Most attacks are also polymorph, and skilled attackers exploit this polymorphism to evade detection.

The obvious solution would be to use an anomaly detection approach, modeling what is *normal* instead than what is *anomalous*. This is surprisingly similar to the earliest conceptions of what an IDS should do [1]. However, while a number of host based anomaly detection systems have been proposed and implemented, both in literature and in practice, network anomaly detection is still an open field for research.

In a previous work [2] we proposed a novel architecture for applying unsupervised learning techniques to a network based IDS. Unsupervised learning techniques are natural candidates for this type of task, but while they have been

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 83–90, 2005.

successfully applied in host based intrusion detection [3], their application to network based systems is still troublesome, mainly due to the problems of input selection, data dimensionality and throughput. In particular, analyzing the payload of TCP/IP packets is a challenging task, which most earlier researches avoided to deal with.

In this paper we will we analyze how our architecture uses the pattern recognition capabilities of a Self Organizing Map (SOM) algorithm [4] in order to solve this problem. By using a SOM, we are able to retain part of the information related to the payload content, characterizing in an unsupervised manner the recurring patterns of packet payloads and "compressing" them into a single byte of information. On most network segments, the traffic belongs to a relatively small number of services and protocols, regularly used, and a good learning algorithm can map it onto a relatively small number of clusters. Our experimental results show that a SOM can successfully learn and recognize these patterns. We also analyze how this unsupervised characterization can help in detecting anomalous traffic payloads. A comparison with various alternative approaches to the problem is also presented.

The remainder of the paper is organized as follows: in section 2 we will describe the proposed architecture for our IDS; in section 3 we will describe how a SOM algorithm can be used for detecting anomalous patterns in payloads; in section 4 we will report on preliminary detection results of the overall architecture; finally, in section 5 we will draw our conclusions and outline some future work.

2 The Proposed Architecture

In a previous work ([2]) we proposed a novel architecture for building a network based anomaly detection IDS, using only unsupervised learning algorithms. These algorithms exhibit properties such as the ability of detecting outliers and of building a model of "normality" without the need of a priori knowledge input; this makes them good candidates for anomaly detection tasks.

We thus reformulated the problem of detecting network anomalies as an instance of the multivariate time sequence outlier detection problem, where the stream of observations is composed of "packets". However, mapping TCP/IP packets as a multivariate time sequence is not straightforward. Each packet has a variable size, and outlier detection algorithms are designed to work on multivariate data with a fixed number of dimensions or "features". The network and transport layer headers can be normalized into a fixed number of features (it is important to note, however, that in the case of connection-oriented protocols the transport layer headers may need inter-correlation in order to be fully deciphered). On the contrary, the data carried by the packet (the payload) cannot be easily translated into a fixed set of features, since each different application layer protocol would require its own set of variables, increasing complexity and decreasing generality. This would also require a full reconstruction of traffic sessions, which would expose the IDS to reconstruction problems, possibly leading to attack windows [5].

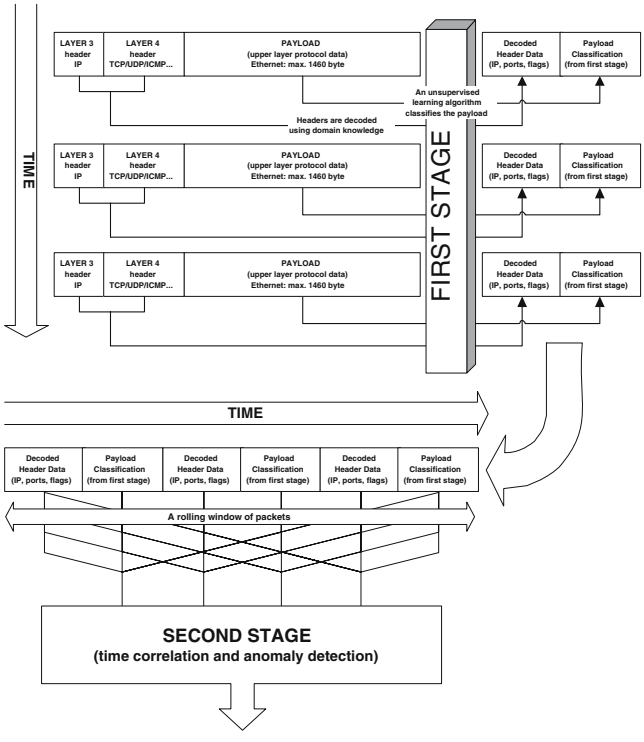


Fig. 1. Architecture of the IDS

Most existing researches on the use of unsupervised learning algorithms for network intrusion detection purposes avoid this problem by discarding the payload and retaining only the information in the packet header (e.g. [6,7,8], the only partial exception being [9], which uses a rule-based algorithm). Ignoring the payload of packets, in our opinion, leads to an unacceptable information loss: most attacks, in fact, are detectable only by analyzing the payload of a packet, not the headers alone. Nevertheless, these algorithms show interesting, albeit obviously limited, intrusion detection properties.

Our approach to the problem consists of a two-tier architecture (shown in Figure 1), which allows us to retain at least part of the information related to the payload content. In the first tier of the system, an unsupervised clustering algorithm operates a basic form of pattern recognition on the payload of the packets, observing one packet payload at a time and "compressing" it into a single byte of information. This classification can be added to the information decoded from the packet header (or to a subset of this information), and passed on to the second tier. On most networks, the traffic belongs to a relatively small number of services and protocols, regularly used, and a good learning algorithm can map it onto a relatively small number of clusters.

The second tier algorithm instead tries to detect anomalies, both in each single packet and in a sequence of packets. It is worth noting that most of

the solutions proposed by previous researchers in order to analyze the sequence of data extracted by the packet headers can be used as a second tier algorithm, complemented by our first tier of unsupervised pattern recognition and clustering.

3 Detecting Patterns in Packet Payloads

In the first tier of our architecture the algorithm receives in input the payload of a TCP or UDP over IP packet, which is a sequence of bytes of variable size (on an Ethernet segment, limited to 1460 bytes). The algorithm must be able to classify such information in a "sensible" way. By sensible we mean that it should preserve three important properties:

1. Preserve as much information as possible about the "similarity" between packets
2. Separate, as much as possible, packets from different protocols in different groups
3. Most importantly, since our final goal is to detect intrusions, separate packets with anomalous or malformed payload from normal packets

"The grouping of similar objects from a given set of inputs" [10] is a typical clustering problem; but it can also be seen as an instance of a pattern recognition problem, where we are trying to characterize the recurring patterns in packet payloads in order to detect anomalies.

We have shown [2] that a Self Organizing Map algorithm [4] is indeed able to sensibly cluster payload data, discovering interesting information in an unsupervised manner, and that it performs much better than a K-means algorithm, or a Principal Direction Divisive Partitioning Algorithm. A previous research showed that neural algorithms can recognize protocols automatically [11], while another paper later independently confirmed our result that the payload of the packets indeed shows some interesting statistical properties [12].

There are multiple reasons for choosing a SOM for this purpose. The algorithm is robust with regard to the choice of the number of classes to divide the data into, and it is also resistant to the presence of outliers in the training data, which is a desirable property. In addition, we have compared various algorithms and shown that the SOM had the best performance trade-off between speed and classification quality.

As it is known, however, the computational complexity of unsupervised learning algorithms scales up steeply with the number of considered features, and the detection capabilities decrease correspondingly (one of the effects of the so called "curse of dimensionality"). There are alternative algorithms for clustering which are much faster in the learning phase than SOM, for example, the well known K-means algorithm is one of the fastest. But during recognition even K-means is not more efficient than a SOM, so we cannot solve this problem by simply choosing a different algorithm.

A traditional approach to the problem would use dimensionality reduction techniques such as dimension scaling [13] or Principal Component Analysis [14].

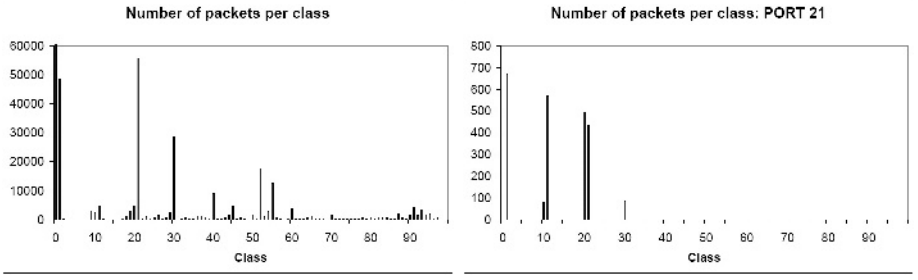


Fig. 2. Comparison between all the traffic and the subset with destination set to port 21/TCP

Table 1. Detection rates and false positive rates for our prototype

Threshold	Detection Rate	False Positive Rate
0.03%	66.7%	0.031 %
0.05%	72.2%	0.055 %
0.08%	77.8%	0.086%
0.09%	88.9%	0.095%

But our experiments demonstrated that they are quite ineffective in this particular situation, since by their nature they tend to "compress" outliers onto normal data, and this is exactly the opposite of what we want to achieve.

Since no alternative solution was viable, we developed various approximate techniques to speed up the SOM algorithm [15]. The throughput of an implementation of the original Kohonen algorithm on common hardware running Linux is on average of 3400 packets per second, which is enough to handle a 10 Mb/s Ethernet network, but insufficient for a 100 MB/s network. Our improved algorithm uses a combination of heuristics in order to reduce the average number of operations to find the best matching neuron, at the expense of precision of matching. The modified algorithm runs at a speed of 10.500 packets/second, which is high enough to handle a normal 100 Mb/s link, with a minimal precision loss which does not impact pattern recognition capabilities. If necessary, performance could be further improved by reducing the number of bytes of the payload considered by the algorithm: it can be shown that this has just minimal impact on the recognition capabilities.

4 Evaluation of Pattern Recognition Capabilities

In order to evaluate the recognition capabilities of the algorithm, we must see if it can usefully characterize traffic payloads for different protocols, and detect anomalous attack payloads from normal payloads. The data used for training and testing the prototype are subsets of the "1999 DARPA IDS Evaluation dataset". In [16] the shortcomings of the DARPA traffic sample sets are analyzed, and

we share many of the author's observations. Thus, we positively validated our results using also smaller dumps collected on our own internal network.

In Figure 2 we present a demonstration of the recognition capabilities of a 10×10 SOM (with hexagonal topology). The network was trained for 10.000 epochs on TCP packet payloads. The histograms represent the number of packets in each of the 100 clusters. For graphical reasons, the scale of y-axis is not the same.

On the left handside, we can see the classification of a whole day of TCP traffic drawn from the dataset. On the right handside, we can see how the network classifies the subset of packets with destination port set to 21/TCP (FTP service command channel). It can be observed how all the packets fall in a narrow group of classes, demonstrating a strong, unsupervised characterization of the protocol.

We also analyzed how the SOM classifies packets from the attacks contained in the DARPA datasets. In Figure 3 we can see that attack packets consistently fall into different classes than normal packets (as an example, we used the packets destined to port 80/TCP).

For the second stage we used a modified version of the unsupervised outlier detection algorithm SmartSifter [17]. We ran the prototype over various days of the 1999 DARPA dataset. The average results are reported in Table 1. The first column contains the sensitivity threshold of the algorithm, and is a statistical predictor of the percentage of data that will be flagged as outliers by the algorithm. As we can see, it is also a good predictor of the false positive rate, if the attack rate is not too high. The prototype is able to reach a 66.7% detection rate

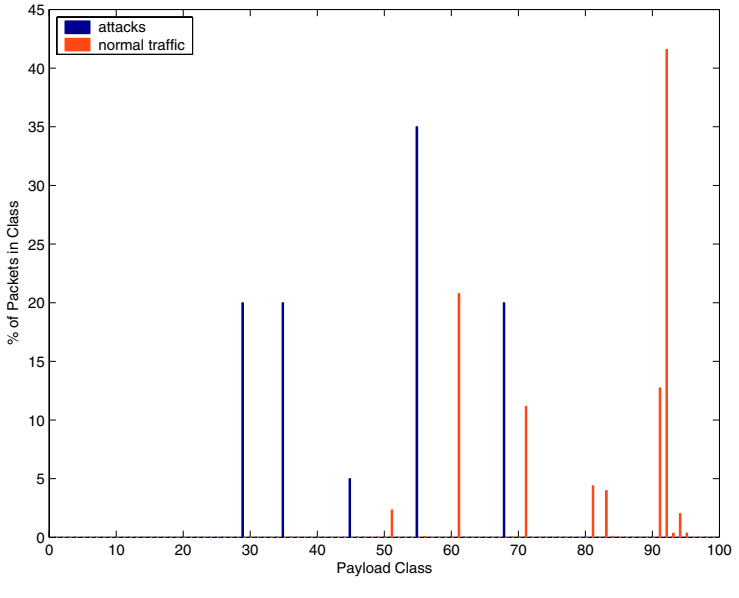


Fig. 3. A comparison between the classification of attack payloads and normal traffic payloads on port 80/TCP

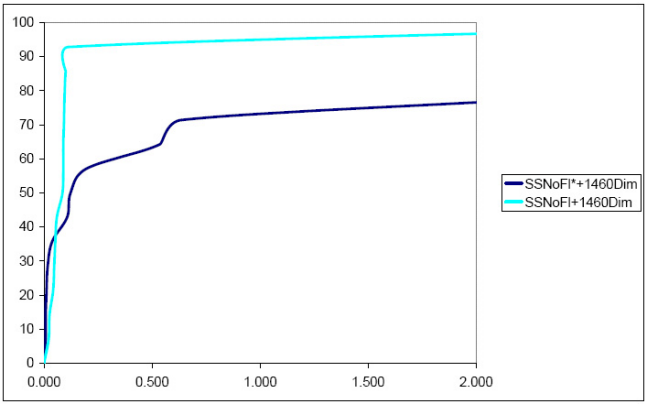


Fig. 4. ROC curves comparing the behavior of Smart Sifter with (lighter) and without (darker) our architecture

with as few as 0.03% false positives. In comparison, in [12] the best overall result leads to the detection of 58.7% of the attacks, with a false positive rate that is between 0.1% and 1%. Our prototype shows thus a better detection rate, with a number of false positives which is between one and two order of magnitudes lower than a comparable system. In Figure 4 we further show how our 2-tier architecture benefits the detection rate by comparing the ROC curves of the system with and without the payload classification stage. The results are clearly superior when the first stage of unsupervised clustering is enabled, proving the usefulness of our approach.

5 Conclusions and Future Work

We have described how we used pattern recognition algorithms in order to build an anomaly based network intrusion detection system. We have described the overall architecture of the system, and shown how the first stage of clustering performs an efficient, unsupervised pattern recognition on packet payloads. The results on the detection rate and false positive rate of the complete system demonstrate that it outperforms a similar, state-of-the-art system by almost an order of magnitude in term of false positive reduction; comparison of ROC curves demonstrates that our approach is indeed the key . Our future work will focus on further false positives reduction, and on the empirical evaluation of the IDS under practical workloads.

Acknowledgments

This work was partially supported by the Italian FIRB Project “Performance evaluation for complex systems”. We need to thank prof. Sergio M. Savaresi for his support and helpful suggestions, and also our student Matteo F. Zazzetta for his invaluable support in software development and lab testing.

References

1. Anderson, J.P.: Computer security threat monitoring and surveillance. Technical report, J. P. Anderson Co., Ft. Washington, Pennsylvania (1980)
2. Zanero, S., Savaresi, S.: Unsupervised learning techniques for an intrusion detection system. In: Proc. of the 14th Symp. on Applied Computing, ACM SAC 2004. (2004)
3. Kruegel, C., Mutz, D., Valeur, F., Vigna, G.: On the detection of anomalous system call arguments. In: Proc. of ESORICS 2003. (2003)
4. Kohonen, T.: Self-Organizing Maps. 3 edn. Springer-Verlag, Berling (2001)
5. Ptacek, T.H., Newsham, T.N.: Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical Report T2R-0Y6, Secure Networks, Calgary, Canada (1998)
6. Mahoney, M., Chan, P.: Detecting novel attacks by identifying anomalous network packet headers. Technical Report CS-2001-2, Florida Institute of Technology (2001)
7. Yeung, D.Y., Chow, C.: Parzen-window network intrusion detectors. In: Proc. of the 16th Int'l Conf. on Pattern Recognition. Volume 4. (2002) 385–388
8. Labib, K., Vemuri, R.: NSOM: A real-time network-based intrusion detection system using self-organizing maps. Technical report, Dept. of Applied Science, University of California, Davis (2002)
9. Mahoney, M.V., Chan, P.K.: Learning rules for anomaly detection of hostile network traffic. In: Proc. of the 3rd IEEE Int'l Conf. on Data Mining. (2003) 601
10. Hartigan, J.A.: Clustering Algorithms. Wiley (1975)
11. Tan, K., Collie, B.: Detection and classification of TCP/IP network services. In: Proc. of the Computer Security Applications Conf. (1997) 99–107
12. Wang, K., Stolfo, S.J.: Anomalous payload-based network intrusion detection. In: RAID Symposium. (2004)
13. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Monographs on Statistics and Applied Probability. Chapman & Hall (1995)
14. Jolliffe, I.T.: Principal Component Analysis. Springer Verlag (1986)
15. Zanero, S.: Improving self organizing map performance for network intrusion detection. In: SDM 2005 Workshop on "Clustering High Dimensional Data and its Applications", submitted for publication. (2004)
16. McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory. ACM Trans. on Information and System Security **3** (2000) 262–294
17. Yamanishi, K., ichi Takeuchi, J., Williams, G.J., Milne, P.: On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In: Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. (2000) 320–324

Conceptual Analysis of Intrusion Alarms

Benjamin Morin and Hervé Debar

France Télécom R&D, Caen, France
{benjamin.morin, herve.debar}@rd.francetelecom.com

Abstract. Security information about information systems provided by current intrusion detection systems (IDS) is spread over numerous similar and fine-grained alerts. Security operators are consequently overwhelmed by alerts whose content is too poor. Alarm correlation techniques are used to reduce the number of alerts and enhance their content. In this paper, we tackle the alert correlation problem as an information retrieval problem in order to make the handling of alert groups easier.

1 Introduction

Security information about information systems provided by current intrusion detection tools is spread over numerous similar and fine-grained alerts. Security operators are consequently overwhelmed by alerts whose content is poor. As suggested by Jakobson and Weissman [7], alert correlation must both reduce the number of alerts and improve their semantics. In intrusion detection, alert correlation techniques can basically be split in two kinds of approaches.

The first kind of approach is a scenario recognition problem. Alarm sequences are matched on-line against the steps of attack scenarios written by experts in a dedicated language ; a high level alert is raised by the correlation system if a scenario is recognized. The approaches proposed by Cuppens and Ortalo [9] as well as Michel and Mé [6] fall into this category.

The second type of approach is a data analysis and clustering problem. Alarms are grouped together according to similarity measures in order to provide the security operator with a synthetic point of view of the security of the information system. The techniques proposed by Valdes and Skinner [11] as well as Cuppens [10] fall into this category. The main critique of this kind of approach is that the alert groups have no description. The number of alerts displayed to the operator is indeed reduced, but the operator still has to investigate the content of the alert groups.

In [8], Julisch proposes a correlation technique based on a variant of a data mining technique called *attribute oriented induction* (AOI). AOI consists in repeatedly replacing alert attributes by more abstract values, according to attribute taxonomies. This approach is interesting because the alert groups are qualified by the abstract values of the taxonomies. However, Julisch's approach has three drawbacks. Firstly, the *hierarchical* structure of the alert groups is restrictive because alerts only belong to one cluster. Secondly, the attribute taxonomies are trees, whose expressive power is limited (a value can only be

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 91–98, 2005.

generalized to another value). Lastly, the alert groups are not constructed incrementally.

In this article, we tackle the alert correlation problem as an information retrieval problem, in order to provide operators with an alert investigation tool for offline use. Information retrieval includes representation, storage, organization and access to information. More specifically, we apply the concept analysis paradigm [5], which is an information retrieval scheme.

In the remainder, we first briefly sketch concept analysis paradigm. Then we define the logic of alerts used to describe the alerts and also to query them. In section 4 we describe a prototype implementation based on a logical file system. Before concluding, we evoke experiences in handling alerts with our approach.

2 Concept Analysis

Existing alert management systems like ACID[1] display alerts in a web-based interface connected to a relational database, where alerts are stored. The security operator submits SQL queries *via* the web interface and the alert management system returns alert subsets which satisfy the query. The number of alerts returned can be huge, so the user has to refine the query in order to select a smaller set of alerts. Unfortunately, there is no simple relation between a variation in the query and the corresponding variation in the answer. The user has to infer an *intensional* modification (*i.e.* a modification of the query) from an *extensional* answer (a set of alerts). Instead of simply displaying the alerts to the operator, the alert management system should also suggest relevant modifications of a query which can be used to reduce the number of answers.

From an information retrieval point of view, this functionality is known as navigation. Information retrieval can be divided in two paradigms : navigation and querying. Querying consists in submitting a request to the information system, which in turn returns the objects that satisfy the query. Navigation consists in storing the information in a classification structure like a tree or a graph. Searching is done by navigating in a classification structure that is often built and maintained manually. We wish to take advantage of the navigation *and* the querying functionality. As shown by Godin *et al.* [4], Formal Concept Analysis (FCA) is a information retrieval paradigm that combines navigation and querying.

Given a *formal context* $K = (\mathcal{O}, \mathcal{A}, \mathcal{I})$ where $\mathcal{O}$ is a finite set of objects, $\mathcal{A}$ is a finite set of attributes and $\mathcal{I}$ is a relation between objects and attributes (*i.e* a subset of $\mathcal{O} \times \mathcal{A}$), formal concept analysis builds a *lattice of concepts* [5]. A *concept* (O, A) is a maximal set of objects $O \subseteq \mathcal{O}$ that share the same properties $A \subseteq \mathcal{A}$. The O part of a concept is called it's *extent* and the A part is called its *intent*. The fundamental theorem of formal concept analysis is that all concepts that can be built on a given formal context forms a complete lattice when it is ordered by set-inclusion of concept extensions.

[1] `http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html`

$$
\begin{array}{lll}
P ::= a \text{ is } v & a \in \mathcal{A}, v \in \mathcal{V} \\
\quad | \quad q & q \in \mathcal{Q} \\
\quad | \quad P \wedge P & \\
\quad | \quad P \vee P & \\
\quad | \quad \neg P &
\end{array}
$$

(a) Alert logic syntax

$$
\begin{array}{ll}
\mu \models a \text{ is } v & \iff v \in \mu[\{a\}] \\
\mu \models q & \iff \mu \models \nu(q) \\
\mu \models P \wedge Q & \iff \mu \models P \text{ and } \mu \models Q \\
\mu \models P \vee Q & \iff \mu \models P \text{ or } \mu \models Q \\
\mu \models \neg P & \iff \mu \not\models P
\end{array}
$$

(b) Alert logic semantics

Fig. 1. Alert logic

The various application domains of FCA bring the need for more sophisticated formal contexts than the mere presence/absence of attributes. Indeed, in the intrusion detection context, we need to handle valued attributes (*e.g.* IP addresses) organized in a taxonomy in order to describe alert groups.

In [1], Ferré proposes a generalization of FCA, called Logical Concept Analysis (LCA). In LCA, sets of attributes are replaced by expressions of an almost arbitrary logic. This property satisfies our requirements.

3 Alert Description Logic

In order to apply LCA to intrusion alerts, we first need to define a description logic for alerts. The logic used to describe alerts is a multi-valued propositional logic, *i.e* a object-attribute-value logic, called $\mathcal{L}_A$. Multi-valued logic have been proved to fulfill the requirements of LCA [2].

The syntax of $\mathcal{L}_A$ is given in Figure 1(a). $\mathcal{A}$ is a set of attributes and $\mathcal{V}$ is a set of values, which are described thereafter. $T = (Q, \nu)$ is a terminology where Q is a set of qualifiers and ν is a mapping between the qualifiers and the formulas of $\mathcal{L}_A$.

Alarm qualifiers can be conceived as aliases for $\mathcal{L}_A$ formula. For example, any formula mapped with the `false_positive` qualifier defines which alert patterns should be considered as false positives.

$\mathcal{L}_A$ is both used as a query language *and* as a description language for alerts. However, alerts are only described by conjunction of attributes. Disjunction, negation and qualifiers are only used for queries.

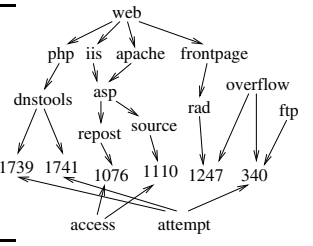
The semantics of $\mathcal{L}_A$ is given in Figure 1(b). In order to give the semantics of $\mathcal{L}_A$, an interpretation is first defined as a relation μ from the attributes $\mathcal{A}$ to the values $\mathcal{V}$ ($\mathcal{M}$ is the set of interpretations). Notation $\mu \models P$ means that interpretation μ satisfies formula. The entailment relation is defined as follows :

$$P \models Q \iff \forall \mu \in \mathcal{M} : \mu \models P \Rightarrow \mu \models Q$$

The taxonomic relations of the attributes domains are considered as axioms of the alert logic. If an attribute value a_1 is more specific than a_0, and an object x has property a_1, then it must also have property a_0. In logical terms this means that $d(x) \models a_1$ must entail $d(x) \models a_0$. This is achieved by considering the taxonomic relationships as axioms of the logic of alerts, *e.g.* $a_1 \models a_0$. For example,

```
1739 WEB-PHP DNSTools administrator
     authentication bypass attempt
1741 WEB-PHP DNSTools access
1076 WEB-IIS repost.asp access
1110 WEB-MISC apache source.asp file
     access
340  FTP EXPLOIT overflow
1247 WEB-FRONTPAGE rad overflow attempt
```

(a) Signatures documentation



(b) Attack taxonomy

Fig. 2. Attack taxonomy

the axiom "VICTIM is `www` $\models$ VICTIM is `web_server`" means that the host whose hostname is `www` is a `web_server`, so the requests which involve `web_server` should include `www` in the answer.

An alert has four attributes types : the attack, the attacker, the victim and a timestamp. Thus, $\mathcal{A} = \{\text{ATTACK}, \text{ATTACKER}, \text{VICTIM}, \text{TIME}\}$. The domains of the attributes are organized in taxonomies. Attribute taxonomies represent background knowledge about the attacks and the monitored environment which is used to enhance the content of the alerts and also to describe the alert groups with abstract values. In this approach, the taxonomies are modeled as directed acyclic graphs (DAGs).

Attack taxonomy. The attack taxonomy is made of attack identifiers and documentation keywords. Attack identifiers are the most specific values of the taxonomy (*i.e.* they are the leaves of the taxonomy). Documentation keywords provide information about the attack properties, such as the type of vulnerable target and the consequences of successful attacks. Relevant keywords are extracted from the plain-text documentation of the attacks by an expert. In our implementation, we used the Snort signature database to build this taxonomy.

Table 2(a) is an excerpt of Snort attack identifiers with their associated documentation ; Figure 2(b) illustrates how they are translated into the taxonomy. In this taxonomy, `cgi`, `php`, `iis`, `apache` and `frontpage` are all kinds of web-like attacks. `dnstools` and `hyperseek` are vulnerable cgi and php scripts, respectively. Numbers in the taxonomy are the signature identifiers; `access` or `attempt` keywords denote the severity of the alerts; `overflow` denotes an attack type (which is independent of the target type).

Victim taxonomy. The victim taxonomy is made of IP addresses, hostnames, host functions and domain names. Victims are referred to with their IP address in IDS alerts, so they are the most specific identifiers of victims in the taxonomy.

Hostnames represent an abstraction of IP addresses. This abstraction is necessary to handle cases where a host is related to more than one IP address. This is the case when a host has several network interfaces, or when the network address translation mechanism (NAT) is used in the monitored environment. For

example, a web server can be referenced by its public IP address in an alert triggered by a sensor located in front of a firewall and by its private IP address in an alert triggered by a sensor inside the DMZ (*DeMilitarized Zone*) of an information system. Hostnames are also more suggestive than IP addresses from a security operator's point of view.

Internal network domain names are an abstraction of hostnames. Hosts (*e.g.* gateways) may belong to several domains. This abstraction level allows the security operator to know if an attack only targets a single host, or if it is a larger-scale phenomenon which involves several hosts of the same subnet.

Hosts functions are also an abstraction of hostnames ; they denote the role of a host in the network (*e.g.* FTP or HTTP server).

Attacker taxonomy. The top of the attacker taxonomy is divided into `external` and `internal` attackers. The internal attackers taxonomy is the same as the victims taxonomy.

Since little information is available about external attackers we use the IANA's[2] database netname field as an abstraction of external IP addresses. The IANA is responsible for the assignments of IP addresses ranges to organisms (ISPs, universities, corporations, etc.). The netname is basically a string that identifies the owner of a IP range. For example, the IP address `80.11.2.166` belongs to the IP range `80.11.2.0-80.11.2.255`, whose netname is `IP2000-ADSL-BAS`. A netname is generally more suggestive for a human operator than an IP address, and it identifies the source as an organism. As such, it offers natural grouping for alerts and differentiates global phenomenons from targeted attacks.

Time taxonomy. We are not interested here in the total order relationship of the time attribute. We want to be able to identify tendencies from an alert set. For instance, we want to know if a given phenomenon only occurs during off-work hours, or during the week-ends. Thus, timestamps are generalized to the hour of the day, and the day of the week. In turn, hours are generalized to `work-hour` or `off-hour` and day of the week are generalized to `week-end` or `week-day`.

4 Implementation

4.1 Prototype

Our implementation consists in storing the alerts in the LISFS logical file system proposed by Padioleau and Ridoux [3]. LISFS provides a generic framework that serves as a basis for any application that requires information retrieval.

In a hierarchical file system, files are located in a single place. Users access the file with their absolute or relative path, whose directories are separated with the `/` character. In LISFS, files are objects and paths are formulas of the logic used to describe the files. The `/` character denotes conjunction, so `propA/propB`

[2] Internet Assigned Numbers Authority, `http://www.iana.org/`

and `propB/propA` contain the same files, *i.e.* those whose description satisfies the `propA` $\wedge$ `propB` logical formula. The `|` and `!` characters respectively denote disjunction and negation in LISFS paths.

Conventional UNIX-like commands are used to query and navigate in the file system. `mkdir` creates a directory, *i.e.* registers a new attribute value. Taxonomic relations like $a_1 \models a_0$ are created using the `mkdir a1/a0` command.

The working directory, denoted by `pwd`, is the current query. `pwd` evolves incrementally, depending on the refinements requested by the user with the `cd` command : `cd p` adds the property `p` to `pwd`, which becomes `pwd` $\wedge$ `p`.

`ls` is used to consult the current place, *i.e* the list of files whose description satisfies the description of the current location. In fact, LISFS does not list *all* the files whose description satisfies the query ; only the files whose description cannot be further refined by any other property are displayed, *plus* the list of relevant properties that can be used to refine the current query. These properties are the navigation links. Moreover, the relevant properties are only the most abstract ones with regard to the taxonomies defined by the user.

StoreAlert	CreateProp
for each d_i **do**	**if** $\neg exists(d_i)$ **do**
CreateProp(d_i)	$D := \{d'_i \mid d_i \models d'_i\}$
done	**for each** $d'_i \in D$ **do**
`cp` m_a $d_1/\ldots/d_n/a$	CreateProp(d'_i)
	done
	`mkdir` $d'_1/\ldots/d'_n$
	done

Fig. 3. Alert storage procedures

In our implementation, each alert produced by an IDS is stored as a file in LISFS. Given an alert identifier a, its description $d(a) = d_1 \wedge \cdots \wedge d_n$ and its content m_a, procedure StoreAlert stores a in the logic file system (see Figure 3). The content of an alert is the raw message produced by an IDS, as well as the traces left by the attack, such as a network packet payload. The identifier of a file is a couple formed with the name of the sensor and a serial number. Command `cp` m_a $d_1/\ldots/d_n/a$ creates a file named a, with description $d_1/\ldots/d_n$ and content m_a. Alarm content can be consulted using the `cat` a command.

The CreateProp procedure adds the missing taxonomic properties entailed by the alert attributes in the file system. Predicate $exists(d_i)$ holds if a property d_i exists in the file system.

For example, if an IP address `192.168.10.80` appears as a victim in an alert, then properties "victim is `192.168.10.80`", "victim is `www`", "victim is `web-server`" and "victim is `dmz`" are added if `192.168.10.80` is a web server located in the DMZ and if these properties have not previously been added.

4.2 Experience with the Approach

The approach has been validated with a corpus containing about 10000 alerts produced by Snort sensors in an operational environment. Alerts are real in the sense that no attack has been simulated.

Submitting `ls -l` at the root of the file system returns 36 navigation links, and no alerts. The following list is an excerpt of the answer :

2182 `acker is int`	8702 `acker is ext`	7033 `vic is dmz`
298 `att is scan`	1 `att is smtp`	6 `att is overflow`
799 `att is snmp`	7039 `att is web`	526 `vic is web-proxy`

The numbers are the number of alerts whose description is subsumed by the associated formula (*e.g.* `att is scan`), *i.e.* the size of the concept extension. `acker`, `att` and `vic` respectively denote the attacker, attack and victim attributes.

Let us suppose that the operator is interested in the alerts whose attack is a web-like attack, and whose victim is a web proxy. The corresponding request is `cd "/att is web/vic is web-proxy"`. When consulting the current place with the `ls` command, the operator sees that among the links proposed by the system, the only one concerning the attacker is `acker is int`, which means that attackers only come from the inside.

Alerts description is subsumed by "ATTACK is `web` ∧ VICTIM is `web-proxy` ∧ ATTACKER is `internal`" are likely to be false positives. Indeed, web proxies forward HTTP requests of the internal users to the outside. However, from the point of view of an IDS monitoring the internal traffic, HTTP requests are addressed to the web proxy. Thus, a suspicious HTTP request would trigger an alert although it is addressed to the outside world, because the IDS thinks the web proxy is the victim of an attack. Since several IDSes signatures are based on the mere presence of known vulnerable web applications in urls (not actual attacks), many false positives are triggered. The operator can map the above alert pattern with the `false_positive` qualifier with the command `mv "/att is web/vic is web-proxy/acker is int" "/false_positive"`. This way, the operator may remove all false positives (`rm false_positive`) or select only alerts which are not known false positives (`cd /!false_positive`).

5 Conclusion

In this paper, we proposed an implicit alert correlation approach based on logical concept analysis. This paradigm allows the operator to handle alert groups in a flexible way, navigate through them and query the system. The alert groups are naturally described with the intent of their corresponding concept. The lattice structure allows intersecting alert groups. Moreover, the lattice construction in incremental.

In order to instantiate the LCA paradigm to the alerts, we proposed a logic of alerts. The vocabulary of the language allows abstract descriptions of the groups because to the alert attributes are structured in taxonomies. Our implementation consists in storing the alerts in the LISFS file system, which is an existing framework for information retrieval-demanding applications.

Experiences with the approach show that the handling of alerts is easier. The current implementation shows limitations in the time required to process alerts as the number of previously stored alerts grows, but newer versions of the logical file systems show very encouraging enhancements in terms of performances.

References

1. S. Ferré and O. Ridoux, A Logical Generalization of Formal Concept Analysis, *International Conference on Conceptual Structures (ICCS 2000)*, 2000.
2. S. Ferré and O. Ridoux, Introduction to Logical Information Systems, *IRISA Research Report RR-4540*, 2002.
3. Y. Padioleau and O. Ridoux, A Logic File System, *Usenix Annual Technical Conference*, 2003.
4. R. Godin, R. Missaoui and A. April, Experimental comparison of navigation in a Galois Lattice with conventional information retrieval methods. *International Journal of Man-Machine Studies*, 38 (5), pp. 747–767, 1993.
5. B. Ganter and R. Wille, Formal Concept Analysis - Mathematical Fundations, *Springer*, 1999.
6. C. Michel and L. Mé, Adele: An Attack Description Language For Knowledge-Based Intrusion Detection, *Proceedings of the 16th International Conference on Information Security (IFIP/SEC 2001)*, 2001.
7. G. Jakobson and M. D. Weissman, "Alarm correlation", *IEEE Network Magazine*, 7(6), pages 52–60, 1993.
8. K. Julisch, "Mining Alarm Clusters to Improve Alarm Handling Efficiency", *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC 01)*, December 2001.
9. F. Cuppens, R. Ortalo, "LAMBDA: A Language to Model a Database for Detection of Attacks", *Third International Workshop on the Recent Advances in Intrusion Detection (RAID'00)*, H. Debar, L. Mé and F. Wu editors, LNCS 1907, 2000.
10. F. Cuppens, "Managing Alerts in Multi-Intrusion Detection Environment", *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC 01)*, 2001.
11. A. Valdes and K. Skinner, "Probabilistic Alert Correlation", *Proceedings of the 4th Recent Advances in Intrusion Detection (RAID2001)*, W. Lee, L. Mé and A. Wespi editors, LNCS 2212, pages 85–103, 2001.

Simplification of Fan-Meshes Models for Fast Rendering of Large 3D Point-Sampled Scenes

Xiaotian Yan, Fang Meng, and Hongbin Zha

National Laboratory on Machine Perception
Peking University, Beijing, P.R. China
{yanxt, mengfang, zha}@cis.pku.edu.cn

Abstract. Fan-Meshes (FM) are a kind of geometrical primitives for generating 3D model or scene descriptions that are able to preserve both local geometrical details and topological structures. In this paper, we propose an efficient simplification algorithm for the FM models to achieve fast post-processing and rendering of large models or scenes. Given a global error tolerance for the surface approximation, the algorithm can find an approximately minimal set of FMs that covers the whole model surfaces. As compared with splat-based methods, the FM description has a large simplification rate under the same surface fitting error measurement.

1 Introduction

3D models and scene representations are of great potentiality to be integrated in Augmented Reality (AR) applications to enhance both realism in the environment rendering and interactive effects. To make them really useful in practical tasks, however, the geometrical descriptions of the models and representations should meet the following requirements: (1) The 3D datasets have to be simplified enough to allow for real-time rendering operations on general computing platforms. (2) Local editing or retargeting can be performed easily on the model surfaces. (3) The data structures of the descriptions are suitable for interactive and progressive transmission.

With advancements in 3D scanning technologies, we can obtain high-quality sampling points for large scenes much more conveniently than before [1]. Since the size of current available scanning data increases rapidly and the topological information becomes very complex, it is a hard work to perform post-processing on these huge sampling point datasets. In general, there are two approaches for the 3D data processing on original scanning data: (1) mesh-based data processing, where completed mesh models should be generated before the post-processing; (2) point-based processing that can be carried out directly on the sampling points. Traditional 3D data processing on simplification and editing operations are designed based on mesh models [2]. However, it is unpractical in most cases due to the expensive computational costs and storage requirements. In recent years, point-based geometry processing has gained increasing attention as an alternative surface representation, both for efficient rendering and for flexible geometry processing [3]. Since we need not to store and maintain globally consistent topological information during the data processing, it is easy to

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 99 – 106, 2005.

handle huge point-sampled geometry by data partition and integration operations [4]. Until now, some techniques have been developed for the simplification [5], rendering [6], transmission [7] of the point-sampled models. Compared with mesh models, however, the point-based approach has difficulty in realizing local editing on the point sets because there has no any connectivity information kept among the points. Moreover, point-based rendering are mostly implemented based on planar primitives, and hence local geometrical details have to be preserved with great care [8].

To solve the problem, we have presented a new kind of local mesh descriptions, called Fan-Meshes (FMs), which can not only represent local geometrical details but also allow for flexible operations on the models. FMs have advantages for the post-processing of huge sampling point datasets, such as building multi-resolution data structure or interactive transmission. In order to combine FMs into AR applications, in this paper, we propose an efficient simplification and rendering algorithm for models represented by FMs. We can decrease the number of FMs greatly by a re-sampling operation. For a given simplification rate, more accurate geometrical representation will be acquired by FMs than by using splats [10], the most popular primitives in point-based rendering. Experimental results illustrate that our simplification method can improve the usage of FMs in AR applications, and a good tradeoff can be achieved between the running speed and visual effects.

2 Generation of FM

At first, the definition of FM is given in this section. Then we explain in detail the FM's generation process that consists of two steps: organizing and selecting neighbors for a center point, and determining the vertices and region of FM. At last, we present a unified error-based criterion to compare FMs with splats.

2.1 FM's Definition

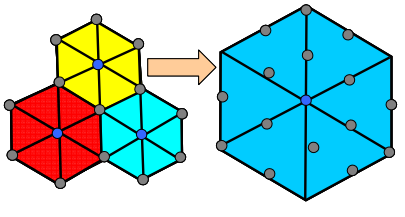


Fig. 1. Left: Original FM; Right: FM we use in this paper

FM is a geometrical primitive we proposed in [9] to reconstruct 3D models and scenes represented by dense scanning point clouds. An original FM consists of three components: a center point, its neighbor points and the connection relationship among these points. As illustrated in the left of Fig. 1, the blue points are center points, the gray ones are their neighbors and six triangles with the same color form a FM. The model surface can be visualized without holes after rendering a large number of FMs with enough overlaps.

The definition of FMs can be modified further to make them adaptive to the shape changes on the model surface. The number of triangles that a FM contains depends on local geometrical properties and its shape depends on the neighbors' distribution. Since original point clouds are very dense and many adjacent FMs may locate nearly on the same planes in large scenes, the FMs will be redundant if we draw them all.

We can use a big FM shown in the right of Fig. 1 to cover all the co-planar points. That is the FM we use in this paper with new properties: First, while all vertices of the original FM are selected form the point cloud, most of the FM's vertices here are not certainly ones on the initial model. Second, in order to get better visual effects, we assume that every FM consists of six triangles and the FM's projection on the least squares plane is a regular hexagon.

2.2 Organizing and Selecting Neighbors

Given an error tolerance E, we must select the largest neighbor which makes the error of FM stay below E for a center point. This step is similar to the splat generation used in [10]. We will discuss the differences between FM and splats in detail in Section 2.4.

The first task is to find the largest neighbor for a center point P_0 under the error toleration E. We estimate the local normal direction $\vec{n}$ by fitting a least squares plane F to P_0 and its k nearest neighbors. For the ith nearest neighbor P_i of P_0, we compute its sign distance h_i and projection distance d_i in the best fitting plane F as

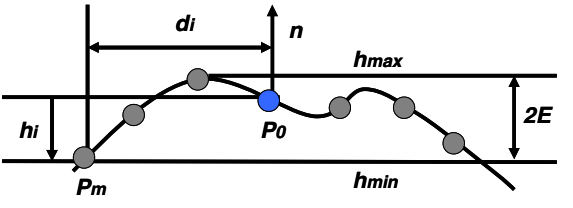


Fig. 2. Find neighbors in a FM

$$h_i = \vec{n} \bullet \left(\overrightarrow{P_i} - \overrightarrow{P_0} \right), \quad (1)$$

$$d_i = \left\| \left(\overrightarrow{P_i} - \overrightarrow{P_0} \right) - h_i \vec{n} \right\| . \quad (2)$$

We rank the neighbors $P_1, P_2,\ldots, P_K$ by their projection distance (Eq. 2). Then we grow the FM by adding other neighbor points and update the interval of their sign distances $[h_{min}, h_{max}]$. The growing stops as soon as the interval between the h_{max} and h_{min} becomes larger than $2E$. As illustrated in Fig. 2, the added neighbors $P_1, P_2,\ldots, P_m$ are the points we need to generate a FM.

2.3 Determining the Vertices and Region of FM

$P_1, P_2,\ldots, P_m$ are the selected neighbors of P_0, and they are ranked by their projection distances. We denote their projections on the best fitting plane F as $Q_1, Q_2,\ldots, Q_m$ respectively. The vertices of FM are $V_1, V_2,\ldots, V_6$ and their projections on F are $R_1, R_2,\ldots, R_6$. We compute the FM that obeys the following three rules: First, the center of FM is P_0 and R_1 is coincident with Q_m. Second, $R_1, R_2,\ldots, R_6$ form a regular hexagon on F. Third, the sum of distances from $P_1, P_2,\ldots, P_m$ to FM is minimal.

It takes too long a time to solve the problem by a least squares method. Here we present a fast algorithm to generate FM by an approximation method. Since the locations of $R_1, R_2,\ldots, R_6$ can be computed by the first and second rules, the vertex V_j locates in the line that is perpendicular to F and through R_j ($j=1, 2,\ldots, 6$). An example of computing the vertex V_3 is shown in Fig. 3. First, we form a coordinate system that centers at P_0, and we assume that the coordinates of R_3 are $(1, 0, 0)$. Then we find a point set $\{S_1\}$ of the neighbors whose projections fall into triangle $P_0R_3R_4$. The center of this point set is denoted as $P_a\ (x, y, z)$ and its projection on F is $Q_a\ (x, y, 0)$. Lines P_0P_a and P_0Q_a have points of intersection with plane $V_3R_3R_4$ respectively. The red line segment in Fig. 3 connects the two intersection points together, and its length is

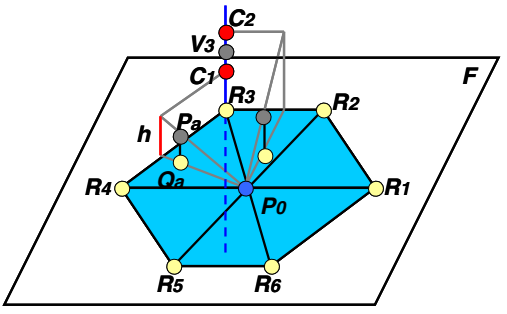


Fig. 3. Determining the vertices

$$h = \sqrt{3}z/(\sqrt{3}x + y) \ . \quad (3)$$

We mark a red point C_1 at line R_3V_3 so that the length of R_3C_1 is equal to h. Then the sum of distances from neighbors in set $\{S_1\}$ to plane $P_0P_aC_1$ is minimal. With a new point set $\{S_2\}$ of the neighbors whose projections fall into triangle $P_0R_2R_3$, we can compute another red point C_2 at line R_3V_3. The vertex V_3 locates in the middle of line segment C_1C_2. We calculate the other five vertices similarly and form the final FM. By this method, the generated FM makes the sum of distances from all the neighbors to it approximately minimal.

In order to perform the simplification process, we have to define an coverage measurement for each FM, which depends on two factors: 1) the area S_a it covers on the surface; 2) the number of points it overlaps.

The area of each sample point is computed first. We define that the area of P_0 is the area of the FM's projection hexagon on F. Since the projection hexagon is regular and its edge length is equal to the longest projection distance d_m, the area of P_0 is

$$A(P_0) = 3\sqrt{3}d_m^2/2 \ . \quad (4)$$

Then we sum the areas of the points that overlapped by the FM and consider the sum as the cover region of FM. The cover region of FM is not equal to the area S_a, but it can be used as the coverage measurement which is given by

$$A(FM) = \sum_{k=0}^{k\le m} A(P_k) \ . \quad (5)$$

The value of $A(P_0)$ depends on the local geometrical properties around P_0, and it is close to the area S_a. As the local geometrical properties around P_k and P_0 are similar, the area value $A(P_k)$ is approximate to $A(P_0)$. As show by the following formulae, the cover region $A(FM)$ is in direct proportion both to the area that the FM covers and to the number of points the FM overlaps.

$$A(P_k) \approx A(P_0) \approx S_a \Rightarrow A(FM) \approx m \cdot S_a \ . \quad (6)$$

2.4 Fitting Accuracy Comparison of FM and Splats

When a plane patch is used as a surface primitive for point-sampled surfaces, we can define the sum of distances from all points to the plane as the standard error measurement. Here, we will give a simple comparison of the splat-based method and our FM method in the surface fitting accuracy by using the error measurement.

Assume the same neighbor selection method is used for our method and the splat simplification method proposed in [10]. Then the standard error of FM is different from that of splats even if they cover the same number of sample points. As illustrated in Fig. 4, the blue line L is a circle splat of P_0, and the two green lines are two triangles of FM we generated. The standard error from the left four points to L is (P_0P'+

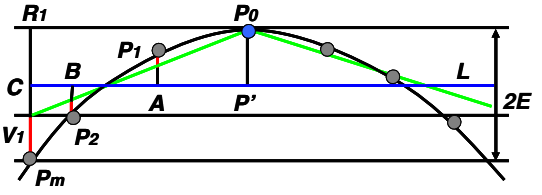


Fig. 4. Standard errors of splats and FM

are presented in Section 4.

$P_1A+ P_2B+ P_mC$), and the error of FM equals to the sum of lengths of the red line segments in Fig. 4 multiplied by $cos(\angle R_1P_0V_1)$, Obviously, the standard error of FM is smaller than that of splats, and the detailed comparison results

3 Post-Processing of FMs and Rendering

Since there are many crossovers among FMs, we can select a subset of the FMs to get a further simplified description. We present two FM simplification algorithms with different purposes here. One aims to have a simple and fast rendering process with consideration on reducing the influence of noisy points at the data boundaries. The other is to get an approximately minimal subset of FMs by using a greedy searching method. The performances of the both methods are shown in the experimental results.

Fast rendering method. Given a sampled point cloud, we divide the points into two sets: points $\{P_c\}$ treated as centers of FMs and points $\{P_d\}$ that are covered by the FMs. For the input point cloud, we set indices for the points in the order of their *XYZ* coordinates. After that, we select the point P_m in the middle of the index queue and put it into $\{P_c\}$. An FM is generated for P_m and we put all points covered by this FM into $\{P_d\}$. The original point set is then divided into two small sets at the location of P_m with P_m and the covered points taken out. The above process is applied to the two sets iteratively until all the points are put into $\{P_c\}$ or $\{P_d\}$.

For a 3D scene with boundaries, the points in the middle zone of the index queue probably distribute in the center region of the scene. By handling the points from center to outside, the boundary points, which usually contain more noises, have little possibility of being selected as the center points of FMs. By doing so, effects of outliers at the boundaries are reduced.

Greedy searching method. The best strategy for the FM simplification is to get a minimal subset of FMs to cover all the sample points. However, it is a NP hard problem and we have to design an algorithm to find an approximation solution with much less costs. In this paper, we use a method based on the greedy searching strategy.

At first, we generate FM_i for every sampled point P_i. Then the areas of the center point $A(P_i)$ and the cover region $A(FM_i)$ are calculated by Eq.(**4**) and (**5**). The inter-coverage relationship is recorded as: 1) the points overlapped by FM_i; 2) the FMs that cover P_i. When the process begins, we select the FM with the maximum cover region and mark the points $\{P_k\}$ it covers as being used. The next step is to update the inter-coverage relationships of remaining FMs. For every point P_j in $\{P_k\}$ we find all the FMs covering it and cut $A(P_j)$ away from the covered regions. Then we repeat the selecting and updating steps until all points are marked as being used.

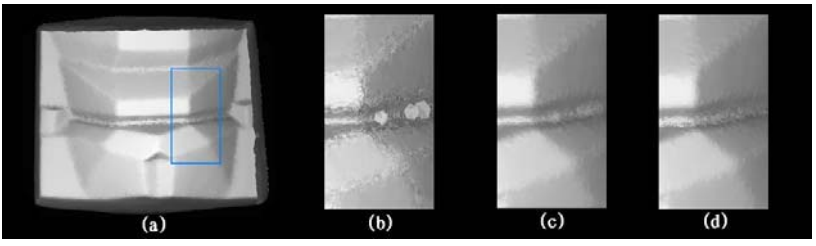


Fig. 5. (a): the simplified mouth model with 5,446 FMs and the region to be zoomed in; (b): fragmentations on the region; (c): the region rendered by the average method; (d): the region rendered by the weighted average method

We need some special techniques to visualize FM models or scenes as there are fragmentations among the FMs. At first, we select the FMs that are at the forefront surface with respect to the viewpoint. These FMs may yet be overlapped by other neighboring FMs, and hence in the corresponding pixel there may be more than one brightness value. Our idea is to use the average of the brightness values to render the pixel. To improve the visual effects, we can set a weight coefficient to every brightness value according to their depths to get a weighted average of the brightness values. The results of our rendering methods are shown in Fig.5.

4 Experimental Results

We apply our algorithm to some 3D models and scenes obtained by using laser scanner. Some performance data that are measured on a Pentium4 2.6GHz CPU with 512MB memory system are given in Table 1. The third column is the Error Tolerance (ET) with respect to the model's bounding box diagonal. The forth and fifth columns are the computing times of the rendering method (Time-R) and the numbers of FMs rendered (FM-R). The sixth and seventh columns are times used in the greedy searching method (Time-S) and the numbers of FMs after simplification (FM-S). The last two columns are the Average Standard Error (ASE) of models obtained by the original splat-based method and our FM method, respectively.

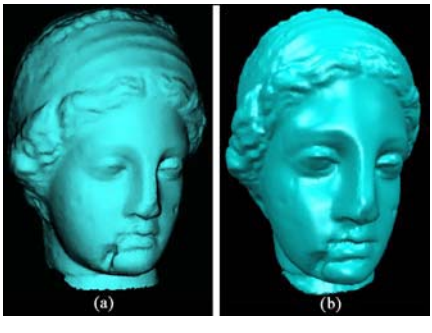


Fig. 6. Simplified Igea models with the same ASE. (a): 28,199 FMs in our method; (b): 29,847 splats in splatting method [10]

We notice that ASE of the splat-based method is bigger than ASE of the FM based method. Therefore, with the same ASE, we can use less FMs than splats to represent the model but get similar visual effects. The comparison is shown in Fig.6.

Table 1. The experimental results of our algorithm

Model	Points	ET (%)	Time-R	FM-R	Time-S	FM-S	ASE of Splats	ASE of FMs
Mouth	23,649	0.03	1.656	5,446	9.14	3,306	0.5325	0.1468
		0.08	1.047	3,021	10.30	1,760	2.3345	0.6791
		0.14	0.857	2,395	11.65	1,337	4.7422	1.3572
		0.56	0.813	1,375	15.38	562	32.930	9.0826
Horse	48,485	0.04	3.969	12,251	18.68	7,252	0.5094	0.1517
		0.12	2.484	6,923	20.45	4,394	2.5961	0.7807
		0.20	2.282	5,447	24.38	3,104	5.5823	1.6191
		0.80	2.000	3,187	32.49	1,394	36.286	9.6502
Desk	103,645	0.01	12.187	24,674	57.69	15,549	0.6276	0.1640
		0.05	7.516	11,213	73.45	5,832	6.2245	2.0355
Building	220,454	0.01	23.922	39,490	143.45	26,085	0.0233	0.0061
		0.02	17.578	28,117	157.81	16,501	0.0548	0.0162

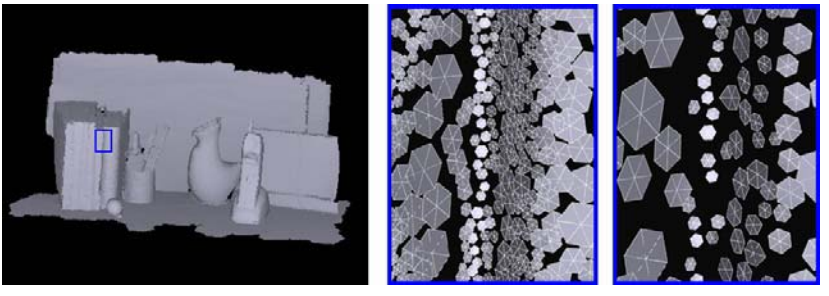


Fig. 7. Distribution of FMs

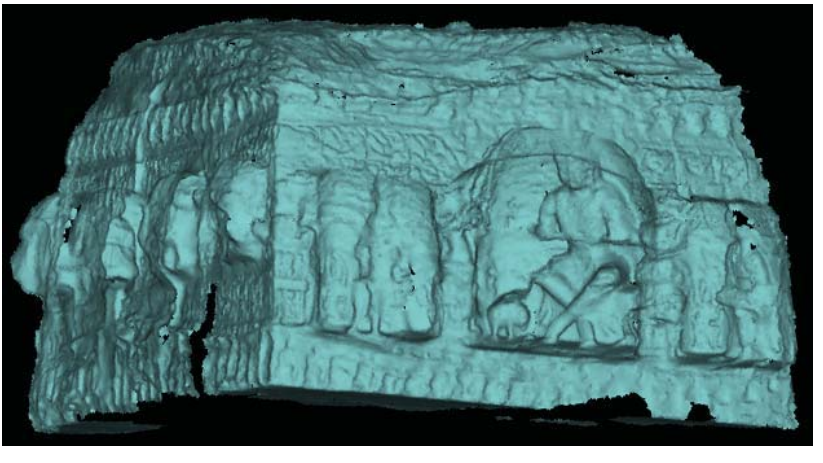


Fig. 8. The YunGang Grotto model with 302,864 FMs. The ASE of this model is 0.032%

Fig. 7 illustrates the distribution of FMs in the results. The left image shows a desk scene with 103645 points, and the blue box indicates the region to be zoomed in. The

middle is the distribution of FMs resulting from the fast rendering method, and the right is the results of the greedy searching method. We find the used FMs are much more sparse but with more uniform distribution.

In order to testify the efficiency of our methods for huge models, we use a dataset of scanning points of the YunGang Grottos in China. It contains 2,981,168 points after registration and it is too large to be held in main memories. Therefore, we cut it into several blocks and perform the simplification on each block. Then we align them together without any special processing for the piece stitching. The whole model is illustrated in Fig. 8, and the overall running time is less than half an hour.

7 Conclusions

In this paper, we proposed an efficient FM-based simplification and rendering algorithm for geometrical description of 3D models or scenes. With our re-sampling operations on FMs, we can decrease the number of the needed FMs greatly, which will improve interactive rendering and transmission performance in augmented reality applications. Since our method is designed based on local geometrical information, it is convenient to perform partial editing and processing on models. Moreover, data structures defined here are easy to integrate into level-of-detail techniques.

Acknowledgement

The work was supported in part by the NSFC Grant (No. 6033010) and NKBRPC (No. 2004CB318000).

References

1. Hu, J., You, S. and Neumann, U., "Approaches to Large-Scale Urban Modeling", *IEEE Computer Graphics and Applications,* Volume 23, Issue 6, 2003, pp. 62-69.
2. Cignoni, P., Montani, C., Rocchini, C., etc., "External Memory Management and Simplification of Huge Meshes", *Visualization and Computer Graphics,* Volume 9, Issue 4, 2003, pp. 525-537.
3. Kobbelt, L. and Botsch, M., "A Survey of Point-Based Techniques in Computer Graphics", *Computers & Graphics,* Volume 28, Issue 6, 2004, pp. 801-814.
4. Meng, F. and Zha, H., "An Easy Viewer for Out-of-core Visualization of Huge Point-Sampled Models", *Proc. 2nd Intel. Sym. on 3DPVT,* 2004, pp. 207-214.
5. Pauly, M., Gross, M. and Kobbelt, L., "Efficient Simplification of Point-Sampled Surfaces", *Proc. IEEE Visualization,* 2002, pp.163-170.
6. Pajarola, R., "Efficient Level-of-Detail for Point Based Rendering", *Proc. IASTED Computer Graphics and Imaging,* 2003.
7. Meng, F. and Zha, H., "Streaming Transmission of Point-Sampled Geometry Based on View-Dependent Level-of-Detail", *Proc. Fourth Intl. Conf. on 3DIM,* 2003, pp.466-473.
8. Kalaiah, A., and Varshney, A., "Modeling and Rendering of Points with Local Geometry", *IEEE Trans. on Visualization and Computer Graphics*, 2002, pp.100-128.
9. Yan, X., Meng, F. and Zha, H., "Fan-Meshes: A Geometric Primitive for Point-based Description of 3D Models and Scenes", *Proc. 2nd Intel. Sym. on 3DPVT*, 2004, pp. 207-214.
10. Wu, J. and Kobbelt, L., "Optimized Sub-Sampling of Point Sets for Surface Splatting", *Computer Graphics Forum (Eurographics 2004),* Volume 23, Issue 3, 2004, pp. 643-652.

Camera Self-localization Using Uncalibrated Images to Observe Prehistoric Paints in a Cave*

Tommaso Gramegna, Grazia Cicirelli, Giovanni Attolico, and Arcangelo Distante

Institute of Intelligent Systems for Automation – C.N.R.
Via Amendola, 122 D-I – 70126 Bari, Italy
{gramegna, grace, attolico, distante}@ba.issia.cnr.it

Abstract. The fruition of archaeological caves, hardly accessible by visitors, can benefit from a mobile vehicle which transmits to users located outside a continuous stream of images of the cave that can be visually integrated with information and data to increase the fruition and understanding of the site. This application requires self-positioning the vehicle with respect to a target. Preserving the cave imposes the use of natural landmarks as reference points, possibly using uncalibrated techniques. We have applied the modified POSIT algorithm (camera pose estimation method using uncalibrated images) to self-position the robot. To account for the difficulty of evaluating natural landmarks in the cave the tests have been made using a photograph of the prehistoric wall paintings of the archeological cave "Grotta dei Cervi". The modified version of the POSIT has been compared with the original formulation using a suitably designed grid target. Therefore the performance of the modified POSIT has been evaluated by computing the position of the robot with respect to the target on the base of feature points automatically identified on the picture of a real painting. The results obtained using the experimental tests in our laboratory are very encouraging for the experimentation in the real environment.

1 Introduction

Earth meanders often contain artistic treasures hardly accessible by visitors for the adverse structural characteristics of the zone and the exigency to preserve these treasures from deterioration and damages. In the south of Italy the walls of the archaeological cave named "Grotta dei Cervi" ("Stag's Cave" *n.d.r*) are rich of red and black paints of hunting, stags and men realized with red ochre and guano of bats. They are among the most remarkable paintings of the European prehistory. In Fig. 1 is illustrated a map of a corridor of the cave hosting four zones, indicated on the map, particularly interesting for the presence of prehistoric wall paintings.

The access to the cave is now denied to unauthorized persons since the exploration of the cave is very hard and a visitor could bring polluting elements in this singular environment. So, the application of a technological solution can allow the remote fruition of the archaeological site with its artistic and historic treasures and, at the same time, can preserve the cave and the safety of the users.

*This work has been developed under project TECSIS supported by grant MIUR n°12905, decreto MIUR 1188, 02.08.2002.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 107 – 114, 2005.

Among several solution, it is currently under development a rover with a sophisticated equipment able to autonomously navigate in the cave and to control the instruments needed for vision, measurements and characterization of the site.

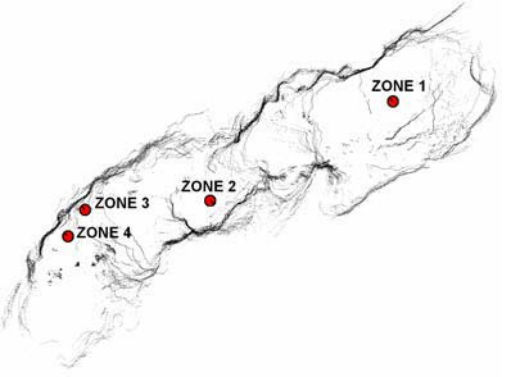


Fig. 1. Map of a corridor of Grotta dei Cervi. Four zones particularly interesting for the presence of prehistoric pictures are indicated

The exigency to preserve the site, avoiding modification of the environment with the installation of external elements, makes necessary to use natural landmarks, in this case wall paintings, as reference points for the rover navigation. A relevant problem in the autonomous navigation of a vehicle is self-localization. This paper deals with the automatic self-positioning of the camera mounted on the vehicle with respect to a predefined target, having the whole target image in the camera field of view.

Several methods to estimate the camera pose can be found in literature [1, 2]. Most of them use landmarks for self-localization [3]. Depending on the knowledge of the 3D target model, model-based [4] or model-free [5] approaches can be used. The POSIT algorithm (Pose from Orthography and Scaling with Iterations), proposed by DeMenthon and Davis [4], is based on geometric considerations that combine perspective and scaled orthographic projections applied to feature points. Being based on a linearized camera model, it is immediate, efficient, works for scenes with a few number of features and has a very low computational cost. It is described in Section 2.

Since the accuracy of 2D features strongly influences the convergence of the method, the pin-hole camera model can not be used for the image representation of the 3D reality. For significant lens distortion, it would be necessary a calibration phase. To avoid this phase, we have modified the original version of POSIT algorithm to use it in real-time applications even with uncalibrated images [6, 7]. The new method, described in Section 3, performs better than the original one (Section 4).

After the validation of the method using a grid target, we have tested the applicability of this method in a complex natural environment such as the archaeological site of Grotta dei Cervi. The only information about this environment come from some photographs of the cave. For this reason, to obtain a 3D target we have created a 3D structure projecting on two orthogonal planes the photograph reproducing some wall paintings that supply a set of feature points.

For the validation of the final results, we have acquired a set of images to evaluate the camera pose for each point of view. The correspondence between the reference 2D

image and the current 2D image is obtained by the Sum of the Absolute Differences (SAD) matching metric introduced in Section 5, through which it is possible to find unambiguous and trackable landmarks in an image. Experimental results relative to the application of the modified POSIT algorithm using a target reproducing prehistoric wall paintings are presented and discussed in Section 6.

2 POSIT Algorithm

The original version of the POSIT algorithm finds the pose of an object from a single image, on the base of the 3D target model in the object coordinate frame of reference. Necessary conditions are the extraction of at least four no-coplanar points and the matching of the extracted features with the corresponding model points.

In a pin-hole camera model, as shown in Fig. 2, a target object, with feature points M_i, is positioned in the camera field of view. The focal length f, the M_i coordinates in the object coordinate frame of reference point M_0 and the image points m_i (x_i,y_i) are all known. The goal is to compute the rotation matrix **R** and the translation vector **T** of the object coordinate frame with respect to the camera coordinate frame.

R is the matrix whose rows are the coordinates of the unit vectors **i**, **j** and **k** of the camera coordinate frame in the object coordinate frame. **T** is the vector $\mathbf{OM_0}$. If M_0 has been chosen to be a visible feature point for which the image is a point m_0, **T** is aligned with vector $\mathbf{Om_0}$ and is equal to $Z_0\mathbf{Om_0}/f$.

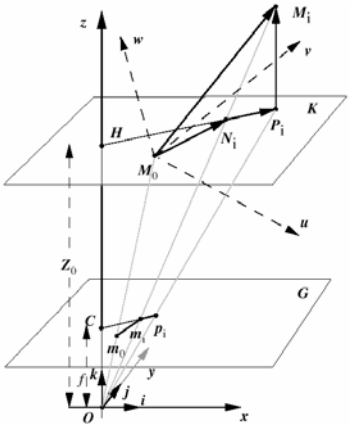


Fig. 2. SOP (p_i) and PP (m_i) of an object point M_i and a reference point M_0

The Scaled Orthographic Projection (SOP) is an approximation of the true Perspective Projection (PP) where all M_i points have the same depth Z_0 as the reference point M_0. If we define the scaling factor of the SOP as the ratio $s=f/Z_0$, the full translation vector $\mathbf{OM_0}$ can be expressed in the following way:

$$\mathbf{OM_0} = \mathbf{Om_0} / s \tag{1}$$

The vector $\mathbf{M_0M_i}$ can be expressed in the following way:

$$\begin{aligned} \mathbf{M_0M_i} \cdot \mathbf{I} &= x_i(1+\varepsilon_i) - x_0 \\ \mathbf{M_0M_i} \cdot \mathbf{J} &= y_i(1+\varepsilon_i) - y_0 \end{aligned} \tag{2}$$

where ε_i, **I** and **J** are defined as $\varepsilon_i = (\mathbf{M_0M_i} \cdot \mathbf{k})/Z_0$, **I**=f **i**/$Z_0$ and **J**=f **j**/Z_0.

If ε_i is known, equations (2) provide a linear system of equations in which the only unknowns are **I** and **J**. Once **I** and **J** have been computed, the scaling factors $s_1 = (\mathbf{I} \cdot \mathbf{I})^{1/2}$, $s_2 = (\mathbf{J} \cdot \mathbf{J})^{1/2}$ and $s = (s_1 + s_2)/2$ are obtained and the unit vectors **i** and **j** derive from the normalization of **I** and **J**.

The POS algorithm finds the pose for which the point M_i has, as SOP, the image point p_i. (ε_i equal to zero). This solution is an approximation because ε_i is not exact. Once **i** and **j** have been computed, a more exact ε_i can be computed in the POSIT algorithm and the equations can be solved again with these better values. By iterating these steps, the method converges to an accurate SOP image and an accurate pose.

3 Pose Recovery with Uncalibrated Images

Equation (1) is valid for a camera perspective approximation, on the basis of the theoretical hypotheses of the POSIT algorithm, but, using uncalibrated real images, this assumption is not valid. The use of a single scaling factor in the definition of **T** is valid only if the image plane has no distortion. The idea on the base of the new version of the method is to use different scaling factors in the definition of **T**.

The three scaling factors s_1, s_2, and s all converge to a single value using the pin hole camera model. This does not happen in the real case for the presence of lens distortion. Since the scaling factors s_1 and s_2 are derived from the normalization of the unit vectors **i** and **j** of the image plane, it is possible to use these scaling factors obtaining a "scaling factor vector" $\mathbf{s_V}$. In this case, **T** can be expressed as follows:

$$\mathbf{T} = \begin{bmatrix} x_0/s_x & y_0/s_y & f/s_z \end{bmatrix} \tag{3}$$

where $\mathbf{s_V} = \begin{bmatrix} s_x & s_y & s_z \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s \end{bmatrix}$.

4 Comparison Between POSIT and Modified POSIT Algorithm

The two versions of the POSIT algorithm have been compared using a 3D grid target, shown in Fig. 3.a, and a Sony DFW-SX900 camera. The extraction of the feature points (black points) has been performed using the Harris corner detection method [8]. For the validation of the final results, nine images have been acquired moving the camera of 10 cm for each acquisition over a grid of size 20x20 [cm] at 155 cm of height. The camera orientation has been set so that the whole target was in the camera field of view. The target was contained at the distance of 98cm from the centre of the observation grid. In the modified version only the x and the y components of **T** have a new formulation. To obtain the reference values for each position, the method of Tsai with an iterative optimization of Lenvenberg-Marquardt [9] has been used.

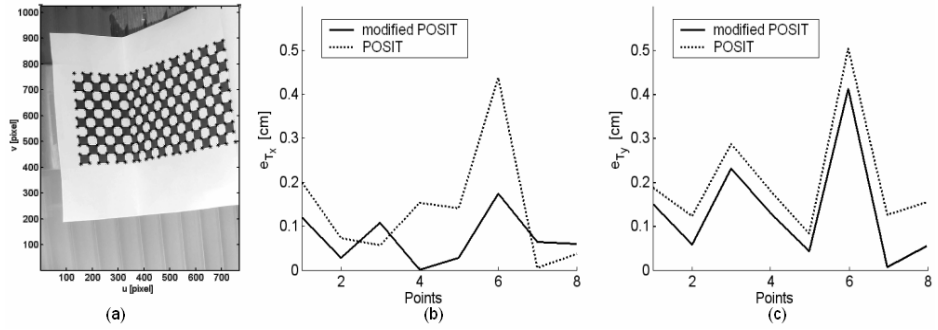


Fig. 3. Grid target (a) and comparison between e_{Tx} (b) and e_{Ty} (c) for the POSIT algorithm and its modified version

Table 1. Pose parameter errors, for the modified version of POSIT, when N is decreased

N	e_x [mm]	e_y [mm]	e_z [mm]	e_θ [°]	e_φ [°]	e_ϕ [°]
231	1.202	1.509	6.415	0.751	0.472	0.488
207	1.359	1.579	6.440	0.895	0.475	0.509
183	1.590	1.615	6.786	1.014	0.518	0.498
159	1.743	1.512	7.608	1.055	0.507	0.490
135	1.943	1.324	8.935	1.100	0.478	0.463
111	1.821	1.174	9.366	1.042	0.450	0.478
89	1.755	1.021	9.932	0.938	0.380	0.444
67	1.577	0.780	10.654	0.907	0.364	0.466
45	1.600	0.543	11.807	0.867	0.315	0.417
22	1.048	0.162	13.825	0.369	0.064	0.284
4	1.100	0.757	16.677	0.132	0.343	0.793

Figures 3.b and 3.c shows the comparison between the x and the y component errors of **T**, indicated with e_{Tx} and e_{Ty}, for the POSIT algorithm and its modified version. It can be observed the improvement of the modified version with respect to the original one.

To avoid calibration patterns, since the convergence of the algorithm requires the extraction of at least four no-coplanar feature points, we have analyzed the pose errors in the central point of the measure grid for the modified version of POSIT (Table 1), when the number of the considered feature points N decreases.

In this case it is possible to use a target with less geometrical constraints. The errors e_i on each pose parameter increase when N decreases. Nonetheless, even in the case of only four no-coplanar points, the difference with respect to the use of a larger N is less than 2 cm.

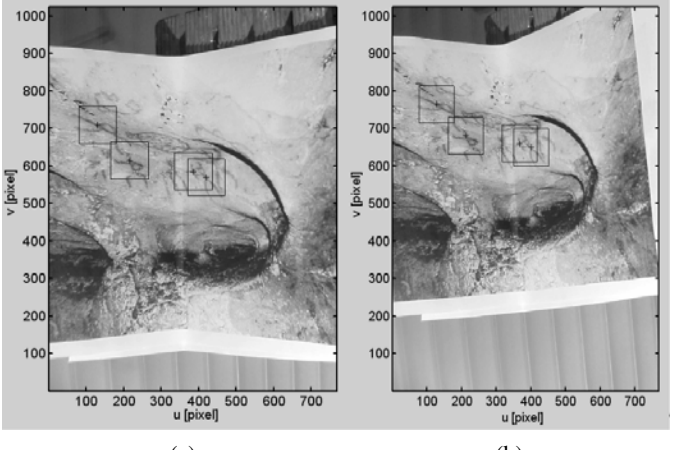
5 Correspondences Between 2D Images

The application of the modified POSIT requires the knowledge of the 3D target model and the extraction of at least four no-coplanar feature points. On the other hand, the only available information about Grotta dei Cervi are some photographs, starting from which

we have obtained a 3D target object by projecting on two orthogonal planes the photograph showed in Fig. 4.

Fig. 4. Photograph of some wall paintings in the Grotta dei Cervi

This photograph shows a wall rich of prehistoric paintings, corresponding to zone 4 indicated in Fig. 1, suitable for the extraction of the necessary features.



(a) (b)

Fig. 5. Example of feature matching between two images

The first image, acquired from the central point of the measure grid described in the previous section, has been used as reference image and the coordinates of the image points corresponding to the 3D model points have been evaluated on this image plane. To extract the right feature points from the other 2D images, corresponding to the feature points of the 3D model, we have used the SAD matching metric that uses the similarity of windows around the candidate points [10]. With this technique, it is

possible to find feature points in an image which are trackable and, at the same time, may not be confused with other possible points.

Fig. 5 shows an example of the matching between the reference image (Fig. 5.a) and another image acquired from another point of the observation grid (Fig. 5.b). The cross markers indicate the matched feature points applying the SAD technique and the squares indicate the windows around each point. The size of the window directly influences the algorithm performance, in terms of results accuracy and computing time, since it delimit the search area for the matchings. We have experimentally set the fitting value for the square size to 101 pixels. As can be observed from the figure, the feature points are extracted from the wall paintings of the cave.

6 Experimental Results

After the validation of the method we have tested the applicability of the modified POSIT for the camera self-localization in a prehistoric cave. Even for this test, the method of Tsai has been used to obtain the reference value for each pose. Once the reference pose values have been estimated, the 3D grid target has been substituted with the 3D target obtained by deforming the photograph of the cave. In this way, it is possible to compare the pose values obtained with the modified POSIT with the reference values obtained with the Tsai method. The experimental results are shown in Table 2. In Table 3 the average μ and the standard deviation σ of the error on each component of the camera pose computed over the set of 9 points of view are shown.

Table 2. Pose parameter errors for each measure point

Point	e_x [mm]	e_y [mm]	e_z [mm]	e_θ [°]	e_φ [°]	e_ϕ [°]
1	0.15	0.81	2.79	1.83	5.57	4.86
2	1.84	0.51	1.44	1.80	5.87	6.01
3	0.25	2.18	10.02	3.43	4.29	0.39
4	0.38	0.57	5.62	2.38	5.83	1.70
5	0.49	0.10	9.13	4.97	0.50	2.53
6	0.01	0.38	7.61	2.26	5.13	5.04
7	0.19	1.06	2.69	1.75	4.76	1.76
8	1.59	0.41	9.36	2.04	4.86	0.69
9	1.68	1.96	14.77	0.05	11.81	1.46

Table 3. Average and standard deviation of the error on each component of the camera pose

	x [mm]	y [mm]	z [mm]	θ [°]	φ [°]	ϕ [°]
$\mu\pm\sigma$	0.70±0.71	0.83±0.64	5.56±3.56	2.43±1.11	4.25±1.94	2.71±2.05

These results are encouraging since the accuracy decreases with larger values of the ratio between the distance camera-target and the object size, as evaluated in a previous validation phase of the POSIT algorithm [6]. For the application in our laboratory, which uses a picture of the target, this ratio is big (it is equal to 25) but, in the real application it is expected to be lower due to the larger dimension of the target (the real paintings of the cave's walls).

7 Conclusions

The use of the modified POSIT for positioning a robot moving in the prehistoric cave of Grotta dei Cervi has been validated. The system provides a stream of images to enable the site fruition by users located outside. This images, and the vehicle position, allow the superposition of visual data to improve fruition and understanding of the site. The modified POSIT improves its performance by using a new formulation of the scaling factor and can work on uncalibrated images and on natural landmarks, automatically identified on the wall paintings to preserve the cave and avoid the installation of external elements. The extraction of the feature points from 2D images, corresponding to feature points on the 3D model of the wall hosting the paintings, has been accomplished using the SAD matching metric.

Experimental tests have been done to both assess the better performance of the modified POSIT with respect to the original formulation and to verify the possibility of reach a satisfactory positioning using natural features identified on the wall paintings. The results obtained in the experimental test in our laboratory are of good omen for the application of this method in this challenging real environment.

References

1. Malis, E.: Survey of vision-based robot control. ENSIETA European Naval Ship Design Short Course, Brest, France (2002)
2. Sugihara, K.: Some location problems for robot navigation using a single camera. Computer Vision Graphics and Image Processing, Vol. 42. (1988) 112-129
3. Betke, M., Gurvits, L.: Mobile Robot Localization Using Landmarks. IEEE Transactions On Robotics And Automation, Vol. 13 (2). (1997) 251-263
4. DeMenthon, D.F., Davis, L.S.: Model-based object pose in 25 lines of code. International Journal of Computer Vision, Vol. 15 (2). (1995) 123-141
5. Basri, R., Rivlin, E., Shimshoni, I.: Visual homing: surfing on the epipole. International Journal of Computer Vision, Vol. 33(2). (1999) 117-137
6. Gramegna, T., Venturino, L., Cicirelli, G., Attolico, G.: Visual servoing based positioning of a mobile robot. Proceedings of the 35th International Symposium on Robotics (2004)
7. Gramegna, T., Venturino, L., Cicirelli, G., Attolico, G., Distante, A.: Optimization of the POSIT algorithm for indoor autonomous navigation. Robotics and Autonomous Systems, Vol. 48(2-3). (2004) 145-162
8. Harris, C.G., Stephens, M.J.: A Combined Corner and Edge Detector. Proceedings of Fourth Alvey Vision Conference, (1988) 147-151
9. Salvi, J., Armangué, X., Batlle, J.: A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation Pattern Recognition, Vol. 35 (7), (2002) 1617-1635
10. Konecny, G., Pape, D.: Correlation Techniques and Devices. Photogrammetric Engineering and Remote Sensing, (1981) 323-333

A Multimodal Perceptual User Interface for Collaborative Environments

Giancarlo Iannizzotto, Francesco La Rosa, Carlo Costanzo, and Pietro Lanzafame

VisiLab, Faculty of Engineering, University of Messina
{ianni, flarosa, ccostanzo, planzafame}@visilab.unime.it
http://visilab.unime.it

Abstract. In this paper a 3D graphics-based remote collaborative environment is introduced. This system is able to provide multiclient and multimedia communication, and exploits a novel multimodal user interaction paradigm based on hand gesture and perceptual user interfaces. The use of machine vision technologies and a user-centered approach produce a highly usable and natural human-computer interface, allowing even untrained users a realistic and relaxing experience for long and demanding tasks. We then note and motivate that such an application can be considered as an Augmented Reality application; according to this view, we describe our platform in terms of long-term usability and comfort of use. The proposed system is expected to be useful in remote interaction with dynamic environments. To illustrate our work, we introduce a proof-of-concept multimodal, bare-hand application and discuss its implementation and the obtained experimental results.

1 Introduction

The field of multiuser, networked Virtual Reality applications is wide and etherogeneous. A number of applications have been developed, each one of them being focused on some specific aspect of graphics [1] [2], communication or collaboration [3], or even portability on most widespread OSes and hardware platforms [4]. Among others, most popular issues in literature have been integrability with well-established Internet applications (such as the Web) [5], mobility of the user [6], portability, and low-traffic communication. In the era of multimodal and multimedia communication, though, the new frontier is represented by the user interaction interface. Perceptual User Interfaces (PUIs) use alternate sensing modalities to replace or complement traditional mouse, keyboard, trackball and joystick input: specialized devices are exploited to provide the user with alternate interaction channels, such as speech, hand gesture, etc. Three classes of technologies can be exploited for PUIs. User-obtrusive technologies are based on sensorized devices such as gloves, jackets, finger- or wrist- mounted sensors [7], which the user must wear before initiating an interaction session. Environment -obtrusive technologies rely on a series of sensors or sensorized devices connected (usually physically attached) to common life objects, such as touch-sensitive flat panels attached to the usual whiteboard or on a desk, which communicate to the computer the needed information about the user interaction. For example, the NTII virtual reality-based communication system [8] is composed by two or more individual stations, located in different

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 115–122, 2005.

places, connected by Internet and each one having its own personal PUI. An almost straightforward alternative to obtrusive technologies is using computer vision to process and analyse the video stream produced by a video camera pointing at the user. Computer vision has been exploited to produce several "demonstration systems", as they are named in [9], for each of the classes listed above.

Indeed, in the past those studies have mainly involved the computer vision and image processing communities, thus producing a plethora of different and often very effective machine vision applications, which in most cases did not address some very important issues related to traditional HCI research, such as usability, ergonomics, compatibility and integration with current applications. On the other hand, in recent years, studies on Augmented Reality (AR) have closely focused on the problems related to human factors such as comfort, long-term usability, user interfaces and perceptual problems, suggesting applications related to advanced visualization for scientific, medical and industrial purposes, entertainment, and soldier augmentation [10].

A Virtual Reality application exploiting vision-based PUI technologies involves combining real and virtual objects in a real environment, running interactively and in real time, registering real and virtual objects with each other. *Such a system matches exactly with the definition of Augmented Reality system* [11]. In this paper, thus, we propose the use of vision-based PUIs to enable advanced interaction with Virtual Reality environments as an Augmented Reality application. We then introduce an Augmented Reality-based communication and collaboration environment, able to provide multiclient and multimedia communication, which exploits a novel multimodal user interaction paradigm based on hand gesture and other perceptual user interfaces. The use of machine vision technologies and a user-centered approach produce a highly usable and natural human-computer interface, allowing even untrained users a realistic and relaxing experience for long and demanding tasks.

The paper is structured as follows: Section 2 describes the presented architecture, motivating the main design options and technology issues; Section 3 illustrates the experimental results obtained from our proof-of-concept application and discusses its implementation details; Section 4 resumes our concluding remarks.

2 System

The system is composed by a number of remote identical units connected each other by a network infrastructure. The users, by means of these units, collaborate in design tasks. Each unit (see Fig. 1) is composed by an entry-level PC, a video projector, a low-intensity infrared spotlight, a network interface, a headset, two videograbbers and two videocameras. Two video streams are thus acquired: one containing the user's face, the other the user's hand gestures and the panel. A metallic support has been built to hold a transparent plexiglas panel coated with a special semi-transparent film for rear projection (see fig. 1), on which the graphical interface is projected through the video projector.

The graphical interface of each unit is splitted in two sections, the first for rendering (and interaction with) 3D Graphical objects and the second to show the remote collaborators we are working with. Each user can *seize* a shared object for editing: when the

Fig. 1. The System

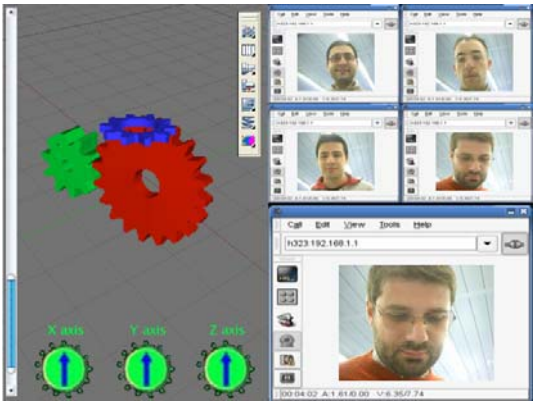


Fig. 2. The Interface

object is seized, the other users can only visualize it and cannot edit it. If an user wants to edit a seized object, he/she can reserve it and will be granted its control as soon as it is again available. Users can create and save a project, add objects to it and "manipulate" the objects as they would do in a real environment with a real object. Once a new project has been created, all the remote users share a common virtual workspace in which different users can integrate or modify some components. A semaphore based policy has been implemented to manage the access and the interaction within the workspace. The second section of panel shows a video-conference environment with its common tools in which each user working on the project is present. All the users at the same time are showed in small windows, while a megnified window shows the user "active" on a specified (clicked) object. Thus, at any time, each user knows who is editing which object.

2.1 Human Computer Interface

The conventional gestures to be used for human-computer interaction should be few, simple and natural.

After a careful evaluation, we choose some of the most common gestures in every day life. For example, the gesture choosen for the common task of *drag&drop* is composecd of the sequence "hand open-close-open" as we would do with real objects to pick them up and release them in the required position. Another functionality, for a simple interaction, should permit to easily select an object or a part of it: for this task we choose the gesture that we use when pointing an object. We obtain the *click event* (the selection) just holding the fingertip upon the object for a few seconds. The *double-click* event is obtained with the sequence " hand open-close-open". *Object resizing* is obtained by selecting an object, extending the thumb and the forefinger and then moving the hand to reduce or magnify the selection. The context menu is opened by extending the thumb. A complete gallery of the gestures we use is shown in Fig. 3.

To get more functionalities, we *augment* our panel with some other graphical tools as a *virtual scrollbar*, placed on the side of the "screen" which we use to zoom in and out the entire workspace. Also, we introduce three *virtual knobs* that allow the user to rotate the workspace around its axes, thus allowing a full 3D vision of the project; the rotation speed is proportional to the rotation angle we virtually impress to the knob.

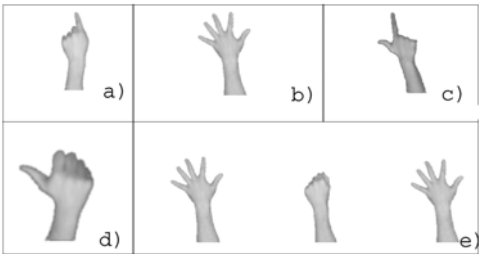


Fig. 3. The complete interaction gestures list: a) point and click, b) rotate, c) resize, d) open menu, e) double-click (or drag & drop)

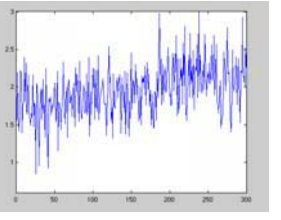


Fig. 4. Standard deviation of the error made tracing free hand an arc of ellipse

The Vision System. The video stream of the user's hand is acquired through a videocamera located under the panel, while the acquisition of the user face is obtained through a videocamera placed in front of her/him. The decision to implement an economical system and thus to use entry-level hardware requires limiting the acquisition resolution to 320x240 for the "panel" and to 176x144 for the "user face" so as to reduce the computational cost while meeting the realtime constraints. The *panel side* of the vision system works in particularly critical lighting conditions: the user hand is lit up mainly by the light beam from the video projector. This introduces a great degree of variability and unpredictability in the lighting of the target (the user hand) of the tracking system. Poor lighting and the need to make the system robust to abrupt background changes due to variations in the image being projected onto the panel make it necessary to have an additional lighting system for the projection surface. This is achieved by using a low-intensity infrared spotlight pointing towards the panel, which increases the overall luminosity but does not affect the projection itself. In this way the user hand appears white in the acquired image against an almost uniform black background. Also, a low-cost infrared filter is placed in front of the videocamera lenses (*panel side*): the effect is to eliminate most of the visible light component, which is mainly represented by the projected images. The overall result is therefore a sufficiently luminous and contrasted image in which the user hand can be seen against a dark, almost uniform background.

The segmentation of the scene, with the user hand moving against a variable background, is performed on each frame acquired by carrying out the following operations in sequence: background subtraction, thresholding, morphological closing and extraction of the connected components.

2.2 Fingers Extraction

Several finger detection systems, available in literature, use signature (an unidimensional representation of an object) analysis with respect to a fixed reference point. Our system analyses the contours of the objects, found in the frame, to check whether an hand is present. We use a contour analysis to understand if the frame contains fingers: for each point $A(x, y)$, the distance from another point B of the contour laying on the perpendicular through A to the contour itself is calculated (see fig. 7). The distances calculated and stored, for each point of the contour, produce an array. For an open hand, the signature is like in Figure 5. Each part of the hand shows some peculiar features that are similar for all users of the application. The figure 5 helps to understand the methods used for the contour analysis. Red section refers to the palm of the hand where the peaks are undoubtedly larger than those related to the fingers, this section is not interesting for fingers detection. Instead the section of array related to a finger (see figure 6) shows some typical features that do not repeat in any other point of the array. In other words, the fingers are distinguished by the presence of a central peak, due to the distances calculated close to the fingertip, and of two flat regions on both sides, due to the distances calculated along the sides of the finger. The green section in Figure 5 refers to the thumb: the peak is lower than in the other fingers but the flat regions on both sides are a little higher than in the other fingers. The other fingers, blue in the figure, are a

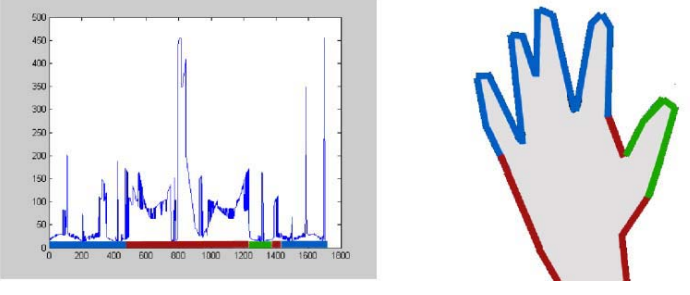


Fig. 5. An example of user hand signature

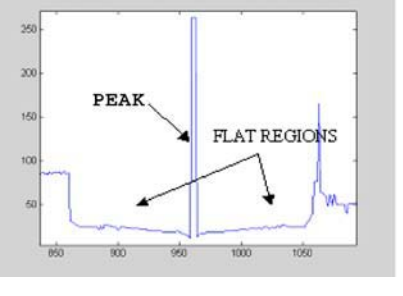


Fig. 6. A Finger in the Signature

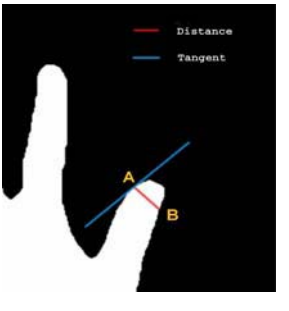


Fig. 7. Contour Analysis Method

little narrower and their peaks are higher. Figure 5 shows that sometimes some peaks can be ambiguous: in these situations a detection based only on the features described above is unadequate or not fully reliable. Therefore, to improve the robustness of the detection, some variations have been introduced. The first approach is to design an algorithm able to select those points candidate to belong to a finger. In this way the range of analysis is reduced to a smaller number of points and consequently the computational cost is lower. The contour of fingers consist of points for which the direction coefficient of the straight line, tangent to the contour and passing for them, changes abruptly. The values of the direction coefficient are stored in an array and those points for which the difference:

$$|arc[i] - arc[i-1]|$$

exceed a fixed threshold are selected. The points for which the direction coefficient changes abruptly can be the fingertips or the *valleys* of the hand. A *valley* is a region of the hand between two fingers: the shape and size of this space make it look like an upside-down finger. To discriminate a valley from a fingertip we determine the sign of the curvature of the contour. Following the contour in the clockwise direction, variations in the sign of the curvature make it possible to distinguish between convex curves (fingers) and concave curves (valleys), whose signs are respectively positive and negative. To do so, we used the method described in [12]. Once the valleys of the hand have been found, the portion of the contour between two of these could be a finger. Finally the signature of the regions classified as fingers are tested to verify the typical configuration of the fingers, i.e. central peak with flat regions on both sides.

Gesture Classification. To discriminate between the active gestures, useful for the interaction, and the neutral postures a classification phase is executed. The distances between the fingertips detected (see section 2.2) and a fixed point (center of gravity of the hand palm) are computed. To obtain the coordinates of this center of gravity a binary image, result of the pre-processing phase, is filtered by a morphological operator (erosion of *FingerWidth* size): we obtain an image in which only the hand palm is present. The center of gravity is a quite stable reference point from which to compute the distance from the fingertips; then, these distances are sorted and used to recognize the fingers extended on the panel. The computed distances are then compared with the values collected during the training phase. Before passing the recognized command to the system, the computed coordinates (fingertip position) need to be corrected because the multimedia video projector and the gray-level videocamera are not orthogonal to the projected surface (panel) and generate a trapezoidal distortion. To do so, we determine the correction parameters using the method described in [12].

3 Experimental Results

The system was tested during and after development by several users for a considerable number of hours with different external lighting conditions. To evaluate the performance of the system a considerable number of tests were carried out and repeated for each of the primitive gestures listed below:***click, double-click, resize, rotate, open menu, drag & drop***.

Table 1. Experimental Results

Command	# Tests	Hits	Near Hits
Click	100	96	//
Double-Click	100	93	//
Resize	100	95	//
Rotate	100	91	//
Open Menu	100	94	//
Drag & Drop	100	97	99

The results obtained are given in Table 1. The percentage of hits in Table 1 indicates the number of times the action was performed correctly, i.e. according to the intention of the user, while the column referring to the percentage of near hits in drag & drop operations comprises both real hits and the number of times the object was released in the wrong place. To produce a quantitative evaluation of the tracker accuracy we compared the output of the system with a ground-truth reference. So we predisposed a test that can meaningfully characterize our system. For this test, we considered an arc of ellipse projected onto the control panel, that must be followed tracing it for its entire length with the fingertip. The measures have been realized asking 10 users to test 5 times the system following free hand the prefixed trajectory, that has been shown on the projection surface, and the system has stored during the tests the coordinates of the output points. An estimation of the whole error (due both to the system and to the accuracy of the user) can be evaluated from the comparison between the curve points coordinates (computed by the system) and the real coordinates of the curve points; carrying out then a statistical analysis on a considerable number of measures we obtained informations about the precision of the system calculating the standard deviations of the errors for each point along the reference trajectory; such errors are expressed in pixel or fractions of pixel. The measures show, particularly in the second half of the abscissas, a defect of accuracy due to the uncertainty of the user. Nevertheless, the extreme naturalness of the system allows to maintain the error under 3 pixels and the analysis of 50 measures carried out shows a medium value of 2 pixels. Fig.4 shows results, along the 300 points of abscissa, of the standard deviation of the error made tracing free hand an arc of ellipse for the test. The increment of the error in the second half of the segment is probably generated from a decay of the attention of the users. Such error is due to the different resolutions of the acquired image and of the projected image. To solve this problem we would need to use an algorithm that allows to obtain a sub-pixel accuracy. This kind of algorithm is usually very computationally intensive thus revealing unsuitable for our purpouses. We therefore decided to keep this error.

4 Conclusions

In this paper we introduced a 3D graphics communication and collaboration environment, able to provide multiclient and multimedia communication, which exploits a novel multimodal user interaction paradigm based on hand gesture and perceptual user interfaces. The use of machine vision technologies and a user-centered approach pro-

duce a highly usable and natural human-computer interface, allowing even untrained users a realistic and relaxing experience for long and demanding tasks. The system has been carefully characterized in terms of accuracy thus allowing for an estimate of its uncertainty and of its usability. Experimental results are shown and discussed.

References

1. Leung, W.H., Goudeaux, K., Panichpapiboon, S., Wang, S.B., Chen, T.: Networked intelligent collaborative environment (netice). In: Proc. of the IEEE Intl. Conf. on Multimedia and Expo., New York (2000)
2. Rich, C., Waters, R.C., Strohecker, C., Schabes, Y., T.Freeman, W., Torrance, M.C., Golding, A.R., Roth, M.: Demonstration of an interactive multimedia environment. IEEE Computer **27** (1994)
3. A.Fersha, J.: Distributed interaction in virtual spaces. In: Proc. Of the 3rd International WS on Distributed Interactive Simulation and Real Time Applications, IEEE CS-Press (1999)
4. Carlsson, C., Hagsand, O.: Dive - a platform for multi-user virtual environments. IEEE Computers and Graphics **17(6)** (1993)
5. : (Smartverse web page) `http://www.smartvr.com`.
6. Ferscha, A.: Workspace awareness in mobile virtual teams. In: WETICE 2000, IEEE Computer Society (2000)
7. Rekimoto, J.: Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In: Proceedings of ISWC'01, Zurich (2001)
8. Towles, H., Chen, W.C., Yang, R., Kum, S.U., Fuchs, H., Kelshikar, N., Mulligan, J., Daniilidis, K., Holden, L., Zeleznik, B., Sadagic, A., Lanier, J.: 3d tele-collaboration over internet2. In: Proceedings of International Workshop on Immersive Telepresence (ITP2002), Juan Les Pins, France (2002)
9. Ye, G., Corso, J., Burschka, D., Hager, D.: Vics: A modular vision-based hci framework. In: Proceedings of ICVS 2003, Graz, Austria (2003)
10. Azuma, R.: A survey of augmented reality. Presence: Teleoperators and Virtual Environments **6** (1997) 355–385
11. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. IEEE Computer Graphics and Applications **21** (2001) 34–47 `http://citeseer.csail.mit.edu/azuma01recent.html`.
12. Costanzo, C., Iannizzotto, G., LaRosa, F.: Virtualboard: Real-time visual gesture recognition for natural human-computer interaction. In: Proc. of the IEEE IPDPS'03, Nice, France (2003)

Robust Correspondenceless 3-D Iris Location for Immersive Environments

Emanuele Trucco, Tom Anderson, Marco Razeto, and Spela Ivekovic

EECE-EPS, Heriot Watt University,
Riccarton, Edinburgh, EH14 4AS, UK
E.Trucco@hw.ac.uk

Abstract. We present a system locating the contour of an iris in space using robust active ellipse search and correspondenceless stereo. Robust iris location is the basis for gaze estimation and tracking, and, as such, an essential module for augmented and virtual reality environments. The system implements a robust active ellipse search based on a multi-scale contour detection model. The search is carried out by a simulated annealing algorithm, guaranteeing excellent performance in spite of heavy occlusions due to blinking, uncontrolled lighting, erratic target motion, and reflections of unpredictable scene elements. Stereo correspondence is avoided altogether by intersecting conjugate epipolar lines with the located ellipses. Experiments on synthetic and real images indicate very good performance of both location and reconstruction modules.

1 Introduction and Motivation

We present a system locating the contour of an iris in space using robust active ellipse search and correspondenceless stereo. Robust iris location is the basis for gaze estimation and tracking, and, as such, an essential module for augmented and virtual reality environments.

Context: immersive videoconferencing. The specific context is our work on the applications of computer vision to immersive videoconferencing [6,7,8,18]. Briefly, a station such as the one in Figure 1 (left) displays real-time, real-size videos of the two remote participants around a virtual table. Videos are acquired by four cameras surrounding the respective, remote plasma screens.

In order to create a visual impression of presence, the remote participants must appear as sitting around the virtual table and must be displayed from the local participant's viewpoint. To this purpose, we warp the incoming video by view synthesis, i.e., we synthesize the correct-viewpoint images [8,18]. This requires two components: real-time viewpoint tracking [19] and dense, accurate stereo disparity maps [6,8]. The latter are hard to achieve given the frequent occlusions created by arm movements, and the wide-baseline stereo geometry typical of immersive VC systems. Interpolation and model-based schemes to produce viable disparity maps have been reported, e.g., in [2,6,7].

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 123–130, 2005.

Tracking the point between the observer's eyes is sufficient to provide the user's instantaneous viewpoint. However gaze, i.e., the direction in which one is looking with respect to the scene, is important for several purposes, including eye contact [3], metadata analysis (e.g., frequency of eye contact with other participants) and affective computing [9].

Monocular gaze is determined by the orientation of the eyeball in space, which is in turn given by the 3-D plane containing the iris contour. This paper concentrates on the problem of locating this plane robustly and accurately.

Fig. 1. Left: an immersive videoconferencing session. Right: active ellipse located around correct iris contour, and segments used to sample intensities.

Related work. Both *invasive* and *non-invasive* iris and pupil location techniques have been reported. Invasive techniques involve the use of devices to be worn or applied, e.g., electrodes, contact lenses and even head-mounted photodiodes or cameras [5]. Non-invasive techniques avoid such solutions but often rely on structured illumination, e.g., Purkinje reflections [5,16,11]. Neither invasive devices nor structured illumination are admissible in our case. We choose not to restrict or control illumination, image quality and iris appearance, which precludes access to well-established techniques for people identification [10,4] relying on well-visible contours or limited eyelid occlusion.

Within immersive videoconferencing, studies have been reported, among others, on eye contact using stereo [3], eye tracking within a 6-camera setup [1] and Hausdorff tracking [19].

The location of the iris contour in space is linked to the problem of locating a conic in space from two perspective projections; closed-form solutions have been reported in [15]. Here, we prefer to exploit a simple model-based constraint to avoid completely stereo correspondence, and reconstruct the iris accurately by calibrated triangulation.

About this paper. In the remainder of this paper, Section 2 sketches the key technical challenges and summarizes our assumptions; Section 3 describes the robust iris detection based on active ellipses, Section 4 describes briefly the correspondenceless stereo module, Section 5 summarizes our experimental assessment of the system, and Section 6 offers some conclusions.

2 Assumptions and Challenges

We intend to estimate the normal to the iris plane and the 3-D iris location in space robustly, repeatably, and without restrictive assumptions or invasive gaze tracking equipment. We assume a stereo pair of cameras imaging a single eye, a setup not atypical in medical environments, biometrics and security. We do not assume special illumination and work with normal room lighting. We do not restrict the position of the iris in the image, nor require that the iris is completely or mostly visible, as assumed in [4].

The challenges are several and not insignificant. We face potentially extensive occlusions by eyelids or eyelashes, regular disappearance of the target due to blinking, frequent erratic target motion, and uncontrolled reflections of unpredictable scene elements and lights (see figures in Section 5). Our solution consists of two modules: robust location of the iris contour (limbus) in each image via active ellipse search, followed by correspondenceless stereo reconstruction of the iris in space. We describe each module in turn.

3 Robust Limbus Detection via Active Ellipses

The input is a monochrome image of a single eye; the output is an ellipse tracing the contour of the iris, illustrated in Figure 2. We suppress corneal reflections and other artefacts introducing distracting, strong contours, with a 10×10 median filter.

Modelling the iris contour. We find the limbus via an optimization in the parameter space of an active ellipse model. The unoccluded portion of the limbus is characterized by a noisy bright (sclera) to dark (iris) intensity transition of varying extent (3 to 12 pixels approximately in our application). We model this transition with two Petrou-Kittler ramp edges [12] at two different spatial scales.

The ellipse is parametrized by its semiaxes, a, b, and centre co-ordinates, O_x, O_y. The axes are assumed aligned with the image axes, as tilt is generally negligible. The cost function extracts intensity profiles along 30 normals to the candidate ellipse, distributed uniformly, as shown in Figure 1 (right) for the correct ellipse. These profiles are, ideally, convolved with two optimal ramp detection filter masks [12] at two different spatial scales. In practice, we are interested only in the filter output at the centre of the normal segments (i.e., at a control point on the ellipse perimeter), so we compute only *one* filtered value per segment. Filtered values are summed over all normals and over both filter sizes to obtain the criterion to optimize, c:

$$c = -\sum_{i=1}^{N}\left(\int_{-w}^{w} S_i(x) f_1(x)dx + \int_{-w}^{w} S_i(x) f_2(x)dx\right), \tag{1}$$

where N is the number of control points, S_i is the intensity profile extracted at the control point i, f_1 and f_2 are the filters at the two different scales, and w is the filter's half-width.

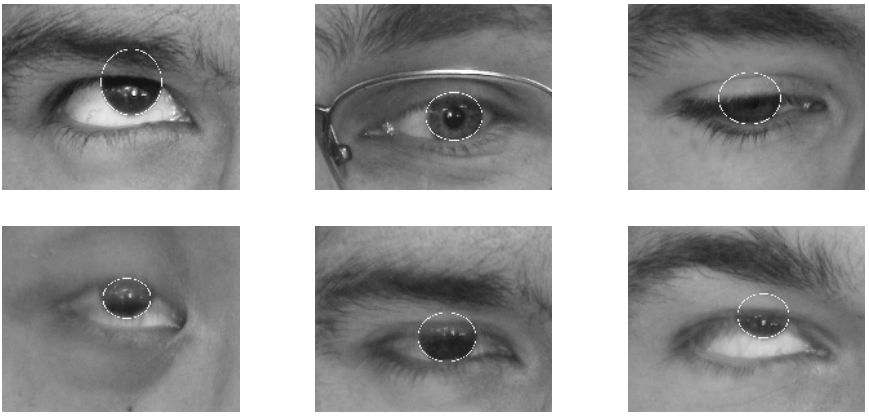


Fig. 2. Examples of iris detection results

Extensive testing identified masks optimal for ramps of width 4 and 10 pixels as responding well to limbus edges in our target images, and poorly to most transitions related to non-iris features (e.g., eyebrows, eyelashes).

Optimization scheme. Deterministic search proved inadequate for our problem, so we analysed various non-deterministic optimizers. We considered standard simulated annealing (henceforth) SA, two SA variations (great deluge, thresholded annealing), and the Girosi-Caprile stochastic optimizer, all reviewed in [14]. We recorded estimation errors in the four ellipse parameters over extensive ranges of variation of the algorithms' parameters, taking care to keep algorithms working in comparable conditions. This work is detailed in [14]. The result indicated standard SA as marginal winner over thresholded annealing.

For reasons of space we can only sketch the SA module. We refer the reader to [13] for details of our implementation, and to Salamon et al. [17] for a full treatment of SA and its practicalities. The active ellipse (i.e., the state vector (a, b, O_x, O_y)) is initialised at the image centre with a default size. The number of ellipses tested is progressively reduced with temperature, from $T_{start} = 500$ to $T_{end} = 1$. These temperature values were decided by sampling the cost function over several images and calculating the relative acceptance ratio, whose desirable value at high temperature is around 50%.

New candidate values for each parameter are generated from a Gaussian distribution centered in the previous value, with standard deviation $\sigma_{new} = R\sigma_{old}$, where R controls the search range, starting from 2 pixels for ellipse centre and 1 pixel for axes lengths and decreasing with an independent annealing schedule. The acceptance rule for new states is the standard Metropolis rule. The annealing schedule affects the move class via the range parameter R:

$$R_{new} = (\frac{1}{\sqrt{t+1}} + 0.3)R_{old}$$

where t is the annealing iteration index (time).

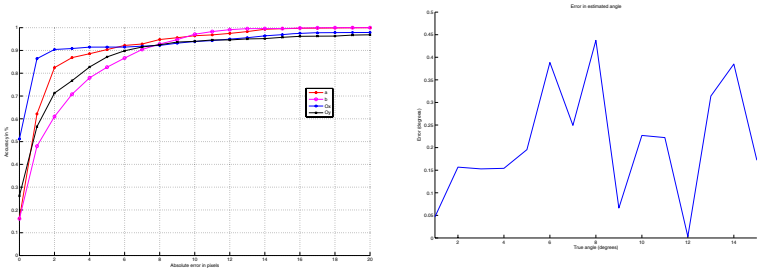


Fig. 3. Top: Monte-Carlo estimates of error probability of absolute error (pixels) for the four ellipse parameters. X axis: absolute error in pixels. Y axis: relative frequencies (probability estimates). Bottom: observed angular difference (degrees) between projections onto the XZ (ground) plane of estimated and true normals to the iris plane.

4 Correspondenceless Stereo

We find corresponding ellipse points without any search by locating the ellipse in both image, then intersecting the ellipses with conjugate epipolar lines. As a single ellipse is located in each image, no ambiguity exists. A circle (modelling the iris) is then fitted to the triangulated 3-D points.

We obtain the epipolar geometry from full calibration, but of course weak calibration (only image correspondences known) would suffice, at least for estimating the orientation of the iris plane in space. As the size of the human iris is very stable across individuals and even races [5], reasonable distance estimates could be achieved even with weakly calibrated cameras.

Figure 4 shows two pairs of images (with no occlusion for clarity), the detected irises, and the bundles of conjugate epipolar lines used for correspondence. The epipolar bundle must be chosen so to guarantee accurate intersections, i.e., the epipolar lines must be as normal to the ellipse as possible at the intersection points. We choose 20 points on the left-image ellipse avoiding the top and bottom arcs, where epipolar lines may approach the ellipse tangent. The points are spaced by 10° intervals along the ellipse, and grouped in two sets symmetric with respect to the vertical ellipse axis.

The 3-D plane best fitting the reconstructed points is found by linear least squares via singular value decomposition. Robust fitting is unnecessary as surely no outliers are present: correspondences are drawn from pre-fitted parametric curves.

5 Experimental Results

Iris detection accuracy. To test the accuracy of iris detection, we used a database of 327 monochrome test images with varying iris occlusion and blur, gaze directions, skin colours and eye shapes, and with and without spectacles. The images were 350 × 270, captured by a digital camera or camcorder with

uncontrolled room lighting. Ground truth was established manually by tracing ellipses following the limbus in each image. We performed 50 runs on each image ($50 \times 327 = 16,350$ runs). The ellipse is initialised always at the image centre, with semiaxes of 40 pixels each (the initial position is immaterial for SA). We computed the difference between estimates of ellipse parameters and the corresponding ground truth values. Examples of detections are shown in Figure 2.

Figure 3 (top) summarizes our analysis, showing, for each ellipse parameter, Monte-Carlo estimates of the cumulative probability of a given error value in pixels (relative frequencies). The graph is obtained by integrating the error histograms plotted for each parameter. For instance, 91.5% of the O_x histogram falls within a 5-pixel tolerance interval, suggesting an *indicative* probability of 91.5% for this accuracy level of the horizontal component. For O_y, this figure is 88%, due to frequent eyelid occlusion.

Correspondenceless stereo accuracy. All stereo tests were run with a MATLAB implementation on a Pentium III PC under Windows. Monochrome, PAL-resolution stereo pairs were acquired with PULNIX PEC3010 cameras and a Matrox Meteor II frame grabber. The stereo pair was calibrated using Tsai's classic procedure [20].

Controlled tests. To establish quantitative ground truth for the iris plane, we fixed a picture of a real iris onto a planar support. The support was rotated through an interval of 15 degrees around a vertical axis in steps of 1 degree. The interval was centered around the head-on direction (iris normal along the Z axis, pointing towards the cameras). For each angle, we estimated the orientation of the iris plane. The cameras were calibrated so that the axis of rotation was the X axis of the world reference frame, allowing consistent comparisons of estimates and ground truth. The interocular distance was 90mm, the focal lengths 12.48 and 10.74mm, and the stand-off distance (from left camera) about 200mm.

Figure 3 (bottom) shows, for each orientation, the angular error on the XZ (ground) plane, defined as the angular difference in degrees between the normals to true and estimated iris plane after projection on the XZ plane. The mean is 0.21°, the standard deviation 0.13°, both below the accuracy with which we could measure ground truth quantities. The full error, i.e., the angular difference between full (not XZ-projected) normals, is larger, in part because our manual positioning system did not guarantee repeatable orientations nor perfectly vertical iris planes, in part because estimated normals did include a small Y component. The mean of the full error was 1.5° and the standard deviation 0.4°, ostensibly still very good results.

Real-eye tests. The above camera setup was used to acquire 20 stereo pairs of real eyes. Examples of images with a superimposed bundle of epipolar lines intersecting the detected ellipse are shown in Figure 4, together with the ellipse arcs fitted in 3-D space. The mean deviation from best-fit planes was 0.1mm, with average standard deviation 0.13mm, and maximum deviation of less than 1mm, suggesting accurate planar reconstruction. We could not measure the accuracy of absolute orientation, for which we rely on the controlled tests.

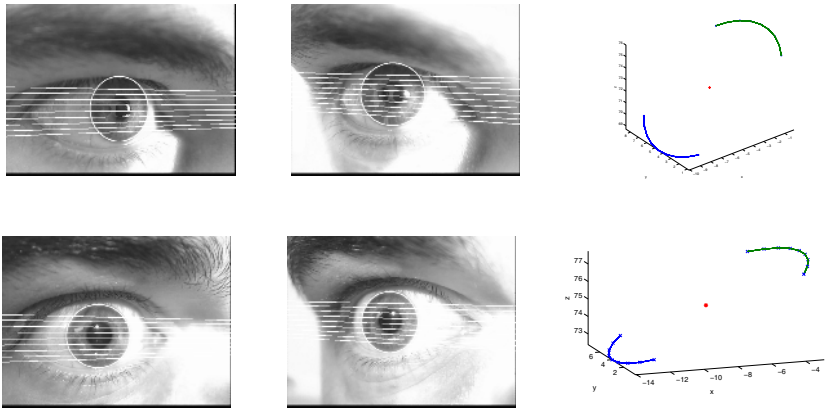


Fig. 4. Located limbus, bundles of conjugate epipolar lines and 3-D reconstruction of 3-D circle arcs for two stereo pairs (one per row)

6 Conclusions

Gaze estimation is an important capability for immersive and collaborative environments, and a crucial component of gaze estimation is 3-D iris location. We have presented a system performing this task reliably. The system implements a robust active ellipse search based on a multi-scale contour detection model. The search is carried out by a simulated annealing algorithm, guaranteeing excellent performance in spite of frequent occlusions due to blinking, uncontrolled lighting, erratic target motion, and reflections of unpredictable scene elements. Stereo correspondence is avoided altogether by intersecting conjugate epipolar lines with the located ellipses. Experiments indicate very good performance of both location and reconstruction modules. Current work is addressing the integration of stereo constraints in the ellipse location.

Acknowledgment

This work was partially supported by an EPSRC CASE scholarship (Anderson) and by OPTOS plc (Razeto).

References

1. Baker, H.H, Malzbender, T., Bhatti, N., Tanguay, D., Sobel, I, Gelb, D., Goss, M., MacCormick, J., Kuasa, K. Cultbertson, W.B.: Computation and performance issues in Coliseum, an immersive videoconferencing system, Proc. 11th ACM Int. Conf. Multimedia (2003), 470–479.
2. Chang, N.L., Zakhor, A.: Constructing a multivalued representation for view synthesis, Int. Journ. of Comp. Vis., **45**:2 (2001), 147–190.

3. Criminisi, A., Shotton, J, Blake, A., Torr, P.: Gaze manipulation for one-to-one teleconferencing, Proc. IEEE Int. Conf. on Comp. Vis., Nice, France, (2003).
4. Daugman, J.: How iris recognition works, IEEE Trans. Circ. Sys. Vid. Tech., **14**:1 (2004).
5. Duchowsky, A.T.: Eye tracking methodology, Springer Verlag (2003).
6. Ivekovic, S., Trucco, E.: Dense wide-baseline disparities from conventional stereo for immersive videoconferencing, Proc. IAPR Int. Conf. Patt. Rec., Cambridge, UK (2004)
7. Isgro', F., Trucco, E., Kauff, P., Schreer, O.: 3-D image processing in the future of immersive media, IEEE Trans. Circ. Sys. Vid. Tech., special issue on immersive telecommunications **14**:3 (2004), 288–303
8. Isgro', F., Trucco, E., Xu, L-Q.: Towards teleconferencing by view synthesis and large-baseline stereo, Proc. IEEE/IAPR Int. Conf. Image Anal. and Processing, Palermo, Italy (2001), 198–203.
9. Kepoor, R., Qi, Y, Picard, R.W.: Fully automatic upper facial action recognition, Int. Wksp. on Analysis and Modelling of Faces and Gestures, IEEE Int. Conf. on Comp. Vis., Nice, France, (2003).
10. Ma, L., Tan, T., Wang, Y., Zhang, D.: Personal identification based on iris texture analysis, IEEE Trans. Patt. Anal. Mach. Intell., **25**:12, (2003).
11. Morimoto, C.H., Koons, D., Amir, A., Flickner, M.: Pupil Detection and Tracking Using Multiple Light Sources, Image and Vision Computing, **18**:4 (2000), 331–335.
12. Petrou, M., Kittler, J.: Optimal edge Detectors for Ramp Edges, IEEE Trans. PAtt. Anal. Mach. Intell., **13**:5 (1991).
13. Razeto, M, Trucco, E.: Robust iris location in close-up images of the eye, Pattern Analysis and Application, to appear, 2005.
14. Richard, S.: Comparative experimental assessment of four optimisation algorithms applied to iris location, MSc Thesis, EECE-EPS, Heriot Watt University, 2004.
15. Schmid, C. Zisserman, A.: The geometry and matching of lines and curves over multiple views, Int. Journ. Comp. Vis. **40**:3 (2000), 199–233.
16. S-W Shih and J Liu, A novel approach to 3-D gaze tracking using stereo cameras, IEEE Trans. Sys. Man Cybern. (Part B), **34**:1 (2004), 234–245
17. Salamon, P., Sibani, P., Frost, R.: Facts, Conjectures and Improvements for Simulated Annealing, SIAM, 2002.
18. Trucco, E., Plakas, K., Brandenburg, N., Kauff, P., Karl, M., Schreer, O.: Real-time disparity analysis for immersive 3-D videoconferencing, Proc. IEEE ICCV Workshop on Video Registration, Vancouver, Canada (2001).
19. Trucco, E., Razeto, M.: Hausdorff iconic matching with applications to 3-D video-conferencing, Proc. WIAMIS, London, UK (2003), 417–422.
20. Trucco, E., Verri, A.: Introductory Techniques for 3-D Computer Vision, Prentice-Hall, 1998.

Input and Display of Hand Drawn Pattern Onto Planar Board of Arbitrary Position and Pose Utilizing a Projector and Two Cameras

Hideo Saito and Hitoshi Ban

Department of Information and Computer Science, Keio University
3-14-1 Hiyoshi Kohoku-ku, Yokohama 223-8522, Japan
saito@ozawa.ics.keio.ac.jp
http://www.ozawa.ics.keio.ac.jp/Saito/index.html

Abstract. In this paper, we propose a system for inputting hand drawn pattern and displaying it on a hand-held planar rectangle board by utilizing two cameras and a projector which are installed above the user. In this system, the cameras and the projector are related by projective geometry. The cameras capture the user's hand-drawing motion. From the captured images, the tip of the user's pen and the corners of the board are tracked for detecting the status of pen's up/down and displaying the drawn pattern at a fixed position on the surface of the board. For demonstrating the validity of the proposed system, we show that the user can draw pattern on the board with arbitrary pose and position, while the pattern is simultaneously displayed on the board in real-time.

1 Introduction

There are a lot of interface devices for using computers. One intuitive interface is using hand-drawn patterns, such as from a pen-tablet interface [1,2]. Hand drawn pattern can also be captured with camera based on computer vision techniques [3,4,5,6,7].

For capturing hand-drawn pattern, we need to detect up/down status of the pen, that is, if the pen is touching on the drawn surface for drawing or leaving for moving to the next drawing stroke. In MEMO-PEN [5], a stress sensor detects the up/down status of the pen. In [7], the length of the stopping time of the pen is used for the detection. In [6,3,4], the pen should actually write the pattern on the surface, so that the camera can detect the locus of the tip of the pen.

In this paper, we propose a system that enables an user to input a hand-drawn pattern on a hand-held surface with arbitrary position and pose, and simultaneously display the input pattern on the surface, by utilizing just two cameras and a projector. In this system, the tip of the pen and the hand-held surface are tracked by two cameras. From the tracked position of the tip of the pen, up/down status of the pen is detected by considering the projective geometry between two cameras and the hand-held surface. The drawn pattern is also captured according to the tracked locus of the tip of the pen, which can be displayed on the surface with arbitrary pose and position by the projector according to the projective geometry between the cameras and the projector. Since the projective geometry among the cameras and the projectors can easily be obtained from the captured images without measuring 3D position/shape of the captured markers/objects, the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 131–138, 2005.

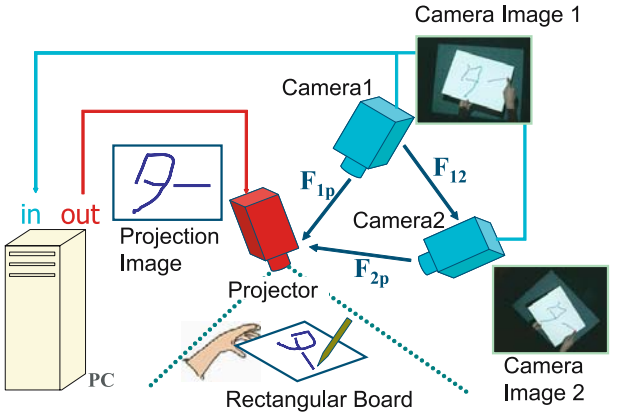


Fig. 1. System configuration

system can easily be built-up without time-consuming calibration of the cameras and the projector with markers and/or reference object with known position and shape.

2 System Setup

Figure 1 shows the overview of the proposed system. By using this system, the user can input arbitrary hand-drawn pattern by drawing with a pen on a board held by his/her hand, while the system simultaneously display the drawn pattern on the board. For realizing such functionalities, the system consists of two cameras for capturing the space of hand-drawing, a projector for displaying the pattern, a rectangular white board held by the user's hand, and a pen for drawing.

The drawing motion is captured by the two cameras. The tip of the pen and the corner of the board are tracked in the captured images. The pen's up/down status is recognized according to the positions of the tip of the pen and the corner of the board

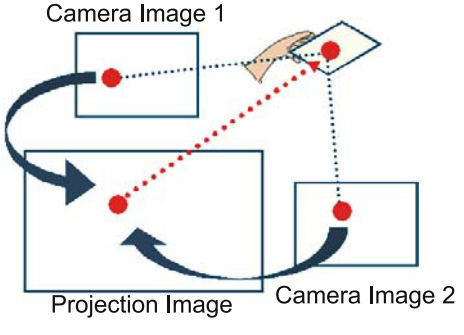


Fig. 2. Computation of Fundamental Matrix

in both images. The locus of the pen is simultanously displayed on the board by the projector.

In this system, projective geometry among the cameras and the projectors should be measured by the following procedure.

The projective geometry are represented by fundamental matrices as $\mathbf{F}_{1p}$,$\mathbf{F}_{2p}$,$\mathbf{F}_{12}$. For estimation of fundamental matrices, 2D-2D correspondences between images captured by the cameras and the image projected by the projector should be detected.

For detecting the corresponding points, an image with a marker point is projected onto the board placed at various positions and poses as shown in Fig. 2. Then the image of the board is captured by the both cameras. Since the 2D position of the marker point in the image projected by the projector is previously known, the 2D-2D correspondence between each camera and the projector can be corrected by detecting the 2D position of the marker point in the image of the board captured by the camera. Using the 2D-2D correspondences, fundamental matrices between each camera and the projector, $\mathbf{F}_{1p}$,$\mathbf{F}_{2p}$, can be computed.

The 2D-2D correspondence between both cameras can also be corrected by the 2D positions of the marking points. The fundamental matrix between both cameras 1 and 2, $\mathbf{F}_{12}$, can be computed from the correspondences.

3 System Behavior

This system takes two images captured with two cameras as input data. The position of the tip of the pen is detected from the input images. For simple and robust performance, we use a pen with a light source on the tip of the pen. The corners of the board are also detected from the input images according to the method in section 3.1.

The pen up/down status can be detected according to the positions of the tip of the pen and the projective geometry of the board, so that the drawn pattern by the user can be captured. At the same time as detecting the positions of the tip of the pen, the projector displays the captured pattern, so that the user can feel as if the user writes the pattern on the board by the pen.

After finishing the hand drawn pattern input/display, the system tracks the corners of the board, so that the captured hand-drawn pattern can always be projected at the fixed position on the board.

Those processes are repeated in the proposed system.

3.1 Detection of Corners of the Board

In this system, the corners of the board are often occluded by the user's hand. Therefore, we find the corners by taking intersection of the line segments of the border of the area of the board.

The procedure is shown in Fig.3. The area of the board is first detected, then the edges are extracted from the board area. The line segments of the edge pixels are are detected by employing the Hough transform. The corners are detected by finding the intersections of the line segments.

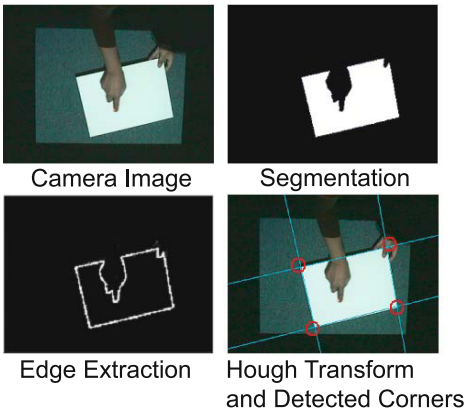


Fig. 3. Flow of corner detection

3.2 Detection of Pen Up/Down Status

For detection of the pen up/down status, projective relationship between two cameras for the board surface, which is represented by homography $\mathbf{H}_{12}$, is considered. For computing the homography $\mathbf{H}_{12}$, the detected corners of the board are corresponded between two camera images as shown in Fig. 4. For determining the corresponding point from detected four corner points, projective relationship between the cameras presented by fundamental matrix $\mathbf{F}_{12}$ is used. As shown in Fig. 4, the closest corner point to the epipolar line is selected as the corresponding point. The use of the fundamental matrix $\mathbf{F}_{12}$ can avoid wrong corresponding points.

Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be the positions of the tip of the pen detected in the captured images. If the tip of the pen is down, $\mathbf{x}_1$ and $\mathbf{x}_2$ should satisfy the equation (1). In other words, the position $\hat{\mathbf{x}}_2$ computed from the detected position $\mathbf{x}_1$ according to the equation (1) should coincide with $\mathbf{x}_2$, if the pen is down. Therefore, we evaluate the distance between $\hat{\mathbf{x}}_2$ and $\mathbf{x}_2$, and then detect if the distance is larger than the threshold th for the pen's up/down detection. as shown in equation (2).

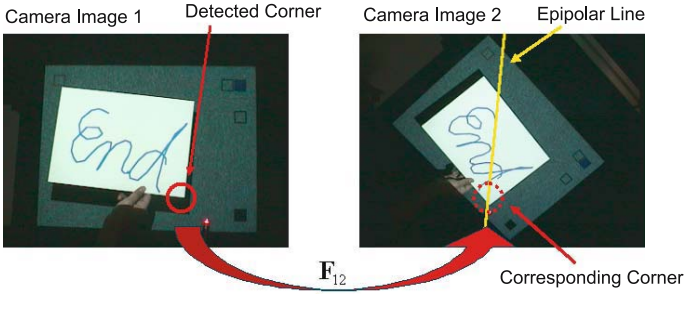


Fig. 4. Calculation of correspondent point

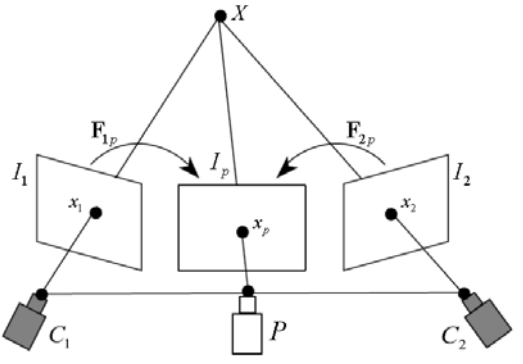


Fig. 5. Making of projection image

$$\tilde{\mathbf{x}}_2 = \mathbf{H}\tilde{\mathbf{x}}_1 \tag{1}$$

$$\begin{cases} \| \hat{\mathbf{x}}_2 - \mathbf{x}_2 \| \geq th : PenUp \\ \| \hat{\mathbf{x}}_2 - \mathbf{x}_2 \| < th : PenDown \end{cases} \tag{2}$$

3.3 Projection of Drawing Pattern

In the proposed system, the drawn pattern is simultaneously displayed on the surface of the board using the projector, when the pen is down.

As shown in Fig.5, I_1 and I_2 represent the images captured by the cameras C_1 and C_2. I_p represents the image projected by the projector P. When the pen is down at point X, let $\mathbf{x}_1$ and $\mathbf{x}_2$ represent the detected position of the tip of the pen in images

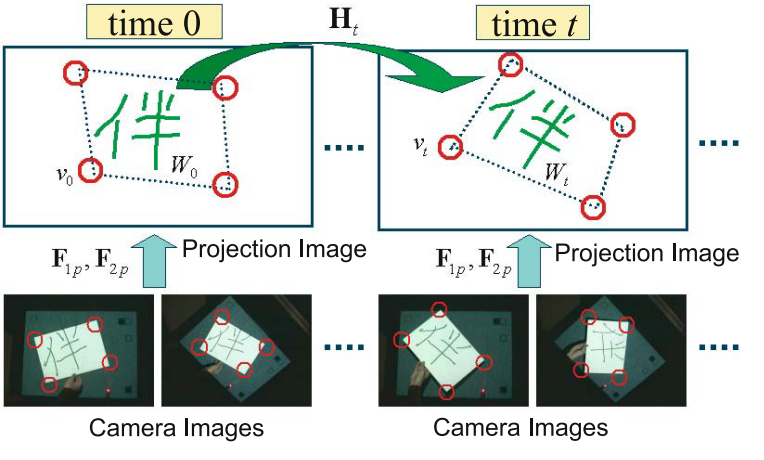


Fig. 6. Warping of hand drawn pattern

I_1 and I_2. Then, two epipolar lines of $\mathbf{x}_1$ and $\mathbf{x}_2$ can be computed on the projection image I_p according to the fundamental matrices $\mathbf{F}_{1p}$ and $\mathbf{F}_{2p}$. The intersection point of the epipolar lines $\mathbf{x}_p$ in the projection image I_p corresponds to the tip of the pen X. Therefore, by projecting the image with a bright point at $\mathbf{x}_p$, the bright point can be displayed at the surface of the board at the point X. In this way, the drawn pattern can be displayed by the projector on the surface of the board.

3.4 Display Pattern After the Drawing

Even after the user finishes the drawing pattern, the system should continue displaying the pattern on the surface of the board. The drawn pattern should be regarded as fixed on the board, the pattern in the projection image needs to be changed according to the motion of the board by user's hand, so that the pattern can be displayed at the fixed position on the board.

Let W_0 represent a projection image of the hand-drawn pattern at time 0. After that, assume that the board moves at time t. If W_0 is still projected at time t, the position and shape of the hand-drawn pattern projected on the board is changed, so the projection image W_0 should be warped to W_t as shown in Fig.6. The warping is performed by shifting all the pixel in W_0 according to the homography $\mathbf{H}_t$ between the board at time 0 and time t.

For computing $\mathbf{H}_t$, the position of the corners detected according to the method in 3.1 are automatically corresponded between time 0 and time, t.

4 Experimental Results

Fig. 7 shows the experimental system. The PC has Pentium4(3.06GHz) and 1024MB memory. The captured images by the cameras are digitized at 320×240 pixels of 24 bit color. The projection image is 1024×768 pixels of 24 bit color. This system runs around 9fps.

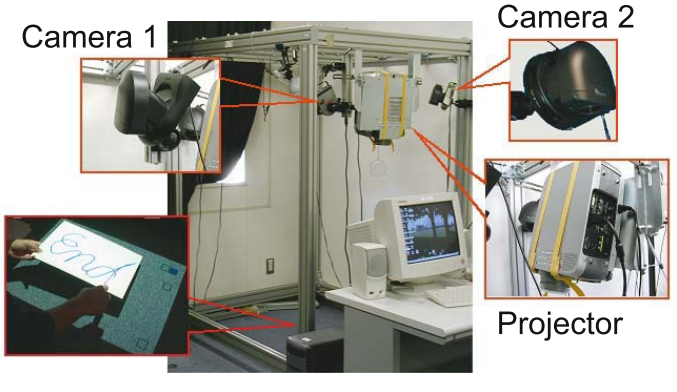


Fig. 7. Experimental system

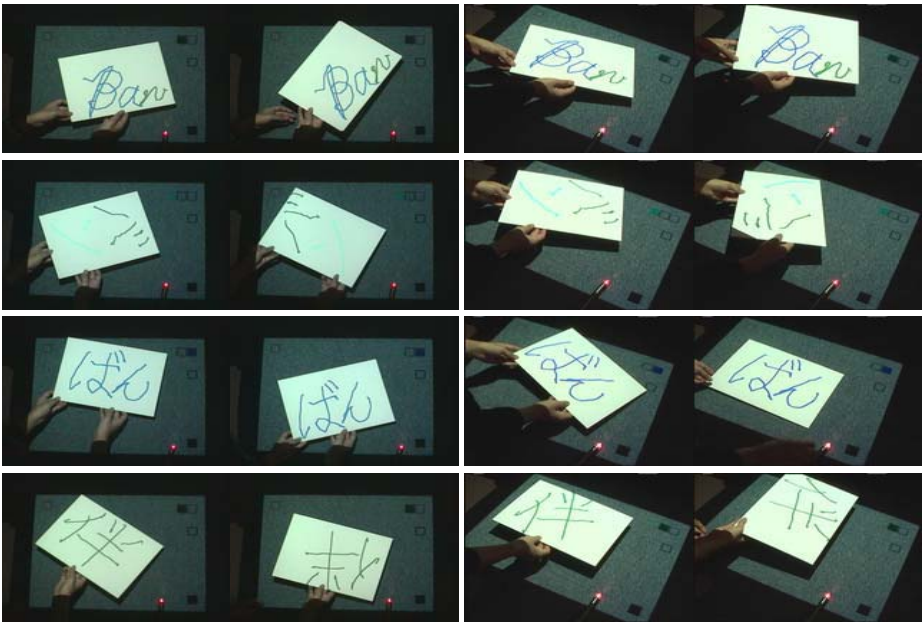


Fig. 8. The appearance when the user draws a pattern using the experimental system

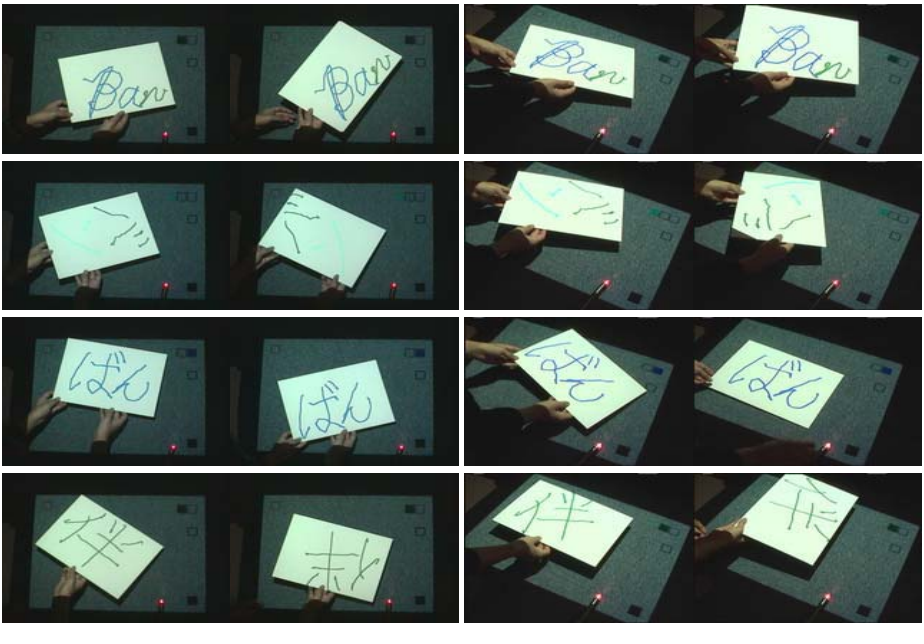


Fig. 9. The appearance when the user moves the board after drawn pattern

Fig. 8 shows the appearance when the user draws a pattern using the experimental system. As shown in this figure, the status of the pen's up/down is detected, so that the hand-drawn pattern can properly be inputted and displayed on the board.

Fig. 9 shows the appearance when the user moves the board after drawn pattern. As shown in this figure, the drawn pattern can be displayed at the fixed position on the board at arbitrary pose and position.

5 Conclusion

In this paper, we propose a system for input and display of hand-drawing pattern on hand-held board by utilizing two cameras and a projector. In the proposed sysytem, the status of the pen's up/down can be detected, so that the drawn pattern can properly be captured and displayed on the surface of the board. The proposed system does not need any 3D information of the cameras, the projector, and the object board, but only needs the projective relationship among them. Therefore, we do not need to perform so-called "strong calibration"

We also show some experimental results for demonstrating the effectiveness of the proposed system.

References

1. Wacom CINTIQ, http://tablet.wacom.co.jp/pro- ducts/cintiq/
2. InkLink, http://www.siibusinessproducts.com/pr- oducts/link-ir-p.html
3. T. Yamasaki and T. Hattori, "A New Data Tablet System for Handwriting Characters and Drawing Based on Image Processing," Proc. IEEE Int. Conf. Systems, Man and Cybernetics, pp.428-431, 1996.
4. H. Bunke, T. Von siebenhal, T. yamasaki, and M. Schenkel, "Online Handwriting Data Acquisition Using Video Camera," Proc. Int. Conf. Document Analysis and Recognition, pp.573-576, 1999.
5. S. Nabeshima, S. Yamamoto, K. Agusa, and T. Taguchi, "Memo-Pen: A New Input Device," Proc. Int. Conf. Human Factors in Computing Systems(CHI), pp.256-257, 1995.
6. M. E. Munich and P. Perona, "Visual Input for Pen-Based Computers," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.24, pp.313-328, 2002.
7. Z. Zhang, Y. Wu, Y. Shan, and S. Shafer, "Visual panel: Virtual mouse keyboard and 3d controller with an ordinary piece of paper," Proc. of ACM Workshop on Perceptive User Interfaces(PUI), 2001.

Computer Vision for Interactive Skewed Video Projection

Alessandro Brazzini and Carlo Colombo

Dipartimento di Sistemi e Informatica,
Via Santa Marta 3, I-50139 Firenze, Italy
colombo@dsi.unifi.it

Abstract. We present an uncalibrated projector-camera system in which the information displayed onto a planar screen can be interactively warped according to an arbitrary planar homography. The user interacts with the system through a laser pointer, whose displacements on the screen plane are captured by the camera, interpreted as mouse drags, and used to control the warping process. Applications of our interactive warping system encompass arbitrary (pan/tilt/screw) keystone correction, visualization of undistorted information for a user being in a general position with respect to the screen (including *virtual anamorphosis* as a special case), and self-shadow avoidance by a nearly-parallel projection.

1 Introduction

Presentation systems based on computer-controlled video projection have recently become a de facto standard in almost every field of professional activity, ranging from business to science and education. Indeed, the digital representation of audio-visual material allows a uniform and highly flexible software treatment of multimedia information, and makes it possible to replace media-specialized display devices as analog VCR's and slide projectors with a single, general purpose digital projector. Projector-camera systems are one of the latest evolutions of the research in the field. These systems use one or more cameras to provide the computer with visual feedback about the presentation scenario, and specifically (1) the information displayed on the screen and/or (2) the human user's activity.

Feedback of type (1) was recently used to automatically compensate for the so called "keystone" deformation. This is a projective deformation of the original display area arising in the presence of misalignments between the projector and the screen planes. Current projectors normally include a hardware keystone correction, but this is typically limited to a 1-dof (tilt) misalignment. The keystone can be represented in the most general way as a planar homography mapping points in the projector plane onto the screen plane, corresponding to a 3-dof misalignment (pan, tilt, screw). To eliminate the effect of the keystone, its associated homography can be estimated and used to suitably pre-deform the image being displayed. Several methods were proposed recently to estimate the keystone homography—without any knowledge of either projector and camera calibration

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 139–146, 2005.

parameters—through the knowledge of the coordinates of some reference points on the screen plane. In particular, in [7], the wall used for display is augmented with four fiducial markers arranged according to a known pattern; similarly, in [1], planar objects with standard shape such as postcards are used; finally, in [8], a blank screen with known shape is used.

Feedback of type (2) was also recently exploited in several fashions in order to improve the way users interact with the system. A general topic of research in this field, actually not limited only to presentation systems, is that of the development of human-computer interaction devices based on the visual interpretation of user motions and gestures, with the purpose of replacing the conventional mouse and keyboard with more natural and effective pointing systems (see [2] for a review). Another research topic, more closely related to the design of camera-projector systems, exploits locating inside the displayed area the laser spot normally used during presentations, so as to infer user intentions and convert them into interface commands. Recent research in this field was focused on using laser pointers so as to simulate mouse clicks, and select interface buttons through a temporal analysis of laser spot changes [4], [5], [8].

In this paper, feedback information of both types (1) and (2) is used so as to design an uncalibrated projector-camera system in which the information displayed onto a planar screen can be interactively warped using a laser pointer according to an arbitrary planar homography. The continuous movements of the laser spot on the screen plane are captured by the camera, interpreted as mouse drags, and used to control the warping process. We demonstrate the usefulness of our interactive warping system (IWS) for several applications, ranging from semi-automatic general keystone correction (not requiring any reference screen points), to visualization of undistorted information for users in a general position with respect to the screen (including *virtual anamorphosis* as a special case), and self-shadow avoidance by a nearly-parallel projection. Experimental results in terms of task completion time and user satisfaction show that the system is suitable for real application scenarios. Video materials are at the page `www.dsi.unifi.it/users/colombo/research/IWS/IWS.zip`.

2 Interactive Warping System

2.1 Overview

Figure 1 shows the main elements of the system. The user (U), projector (P), camera (C), and screen (S) are located in general position with respect to each other. Let I_d be a PC image being displayed, and $I_u(t)$ the same image as perceived by the user at time t. The goal of the system is to cooperate with the user so as to let him gradually change the appearance of $I_u(t)$ to obtain a goal image I_g. If the goal image is equal to I_d, then the user task is to compensate for all the geometric and optical distortions introduced by the system elements. This leads to the application scenarios of *keystone correction* and *purposive misalignment* addressed respectively in subsections 3.1 and 3.2. If $I_g \neq I_d$, then the task is that of *arbitrary viewpoint change*, discussed in subsection 3.3.

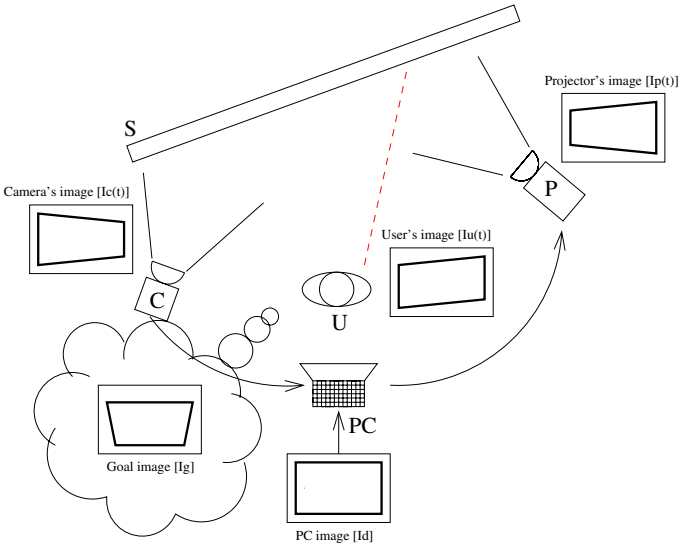


Fig. 1. The elements of the system

2.2 Modeling and Implementation

Mathematically speaking, the different tasks mentioned above can be uniformly described as instances of a general system objective, i.e. to iteratively deform $I_d \in \mathbb{P}^2$ into the projected image $I_p(t)$ according to a time-varying planar homography $H_d(t) : I_d \rightarrow I_p(t)$, until $I_u(t) = I_g$ up to a scale parameter.

The graph of Fig. 2 summarizes the geometric planar-projective relationships (graph edges) between pairs of system elements (graph nodes). Notice that any change to $H_d(t)$ induces a change in all of the images but I_d (and, of course, I_g).

The homography $H_c(t) : I_c(t) \rightarrow I_p(t)$ relates the camera and projector images. If both projector and camera are assumed to be fixed, and to have unknown

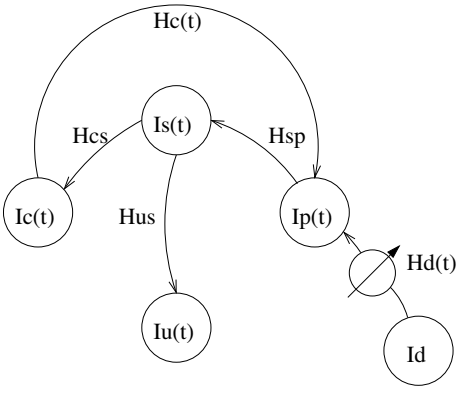


Fig. 2. Geometric relationships between system elements

but constant internal parameters, then $H_c(t)$ is actually a time-invariant transformation H_c, that can be easily computed once and for all at system startup from four or more point correspondences as shown in [3]. The deformation homography $H_d(t)$ is obtained for each t as follows. As the user points at the screen with his laser beam, the position of the laser spot on S is imaged by C at $\mathbf{x}_c(t)$. The point $\mathbf{x}_p(t)$ on the projector's plane can be consequently computed as $\mathbf{x}_p(t) = H_c\,\mathbf{x}_c(t)$. Now, consider the four corners $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ of the image $I_p(t-1)$, and assume without loss of generality that $\mathbf{b}$ is the one closest to $\mathbf{x}_p(t)$. Then, $H_d(t)$ is computed as the transformation leaving unaltered $\mathbf{a}$, $\mathbf{c}$ and $\mathbf{d}$, and replacing $\mathbf{b}$ with $\mathbf{x}_p(t)$. On the user's side, the overall effect is to perceive to be using the laser beam as a means to control the position of $\mathbf{b}$.

The laser spot is localized at each frame on the image plane by a simple red spot detector based on color and intensity thresholding. For the sake of temporal smoothing, the located spot $\mathbf{x}_c(t)$ is obtained as the output of a mobile mean filter with constant gain $\alpha = 0.2$.

3 Applications and Results

3.1 Semi-automatic Keystone Correction

Fig. 3 shows the geometry for keystone correction. In Fig. 3 (left) is depicted the special case of keystone correction where the user is located in a frontoparallel way with respect to the screen. This case is the only case practically addressed in an embedded way by modern video projectors. Similarly, all of the techniques for automatic keystone correction proposed so far in the literature and described in the introduction address only this special case.

The general case of keystone correction, which can be correctly handled by IWS, is shown in Fig. 3 (right). In this case, the user is in general position with respect to the screen, and yet is able to compensate for the distortions induced by the system.

Fig. 4 illustrates the process of keystone correction.The figure also shows the laser spot and one of the four interest regions around the current image corners in which the spot is searched for. Notice that, thanks to its "what you

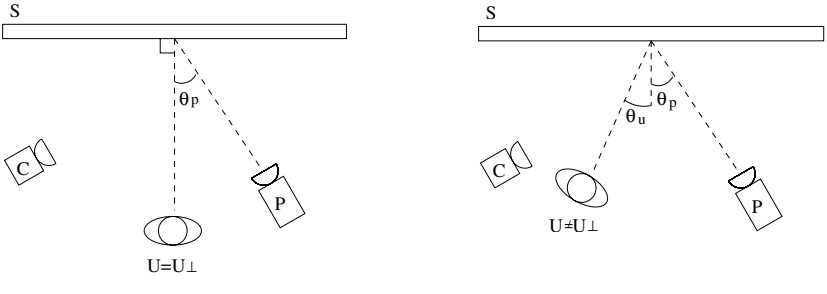


Fig. 3. *(Left)* Standard keystone correction (frontoparallel user). *(Right)* Extended keystone correction (user in general position).

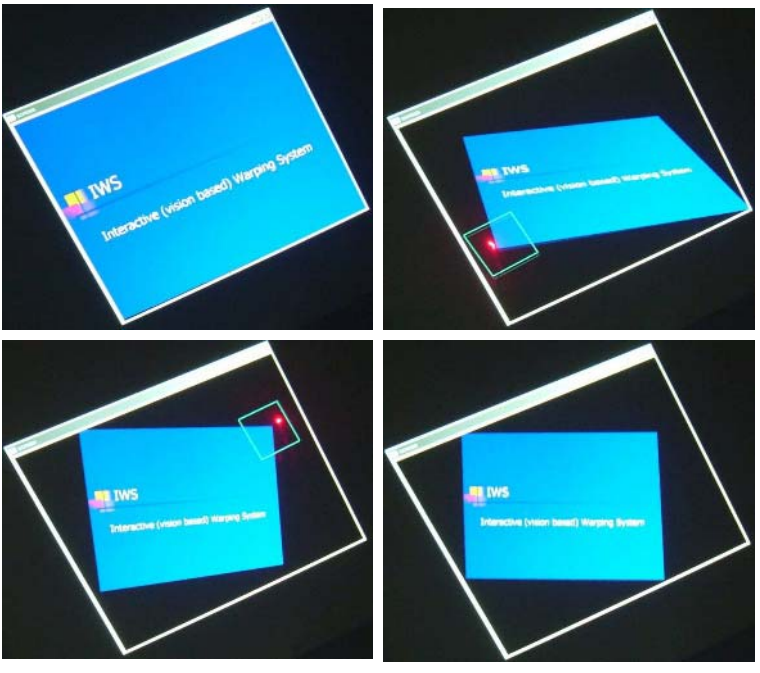


Fig. 4. Semi-automatic keystone correction. *(Upper Left)* Initial view. *(Upper Right), (Lower Left)* Two intermediate phases. *(Lower Right)* Final view.

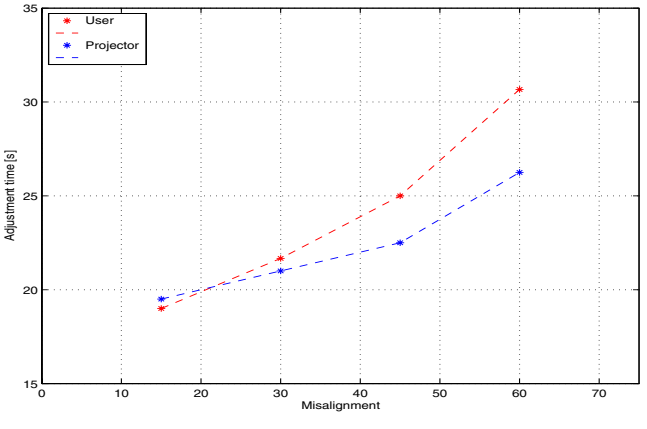


Fig. 5. Task completion time performance for varying projector angles θ_p ($\theta_u = 0$) and user angles ($\theta_p = \pi/6$)

see is what you get" nature, the homography-based approach allows users to cope with general pan/tilt/screw keystone misalignments in a very effective and natural way: The same could not be obtained by three distinct buttons, each controlling a separate dof.

System performance in terms of task completion time for a group of seven different users and a general 3-dof misalignment is shown in Fig. 5. The time required to adjust the keystone slowly and linearly increases as the misalignment angles θ_p (projector) and θ_u (user) with respect to the screen normal increase. A completion time of about 30 s is required for misalignment angles of about 60 degrees. This is not a penalizing performance, also considering the fact that the keystone correction has to be performed at system startup and has not to be repeated very frequently. System performance appears to be slightly worse for user misalignments than for projector misalignments, possibly due to the fact that in the latter case users can often rely on fiducial lines (e.g., the line between the floor and the screen wall) for the completion of their task.

3.2 Purposive Misalignments

The geometric configuration (Fig. 6) for purposive misalignment is similar to that of Fig. 3, but the application domain is different. In fact, the figure shows

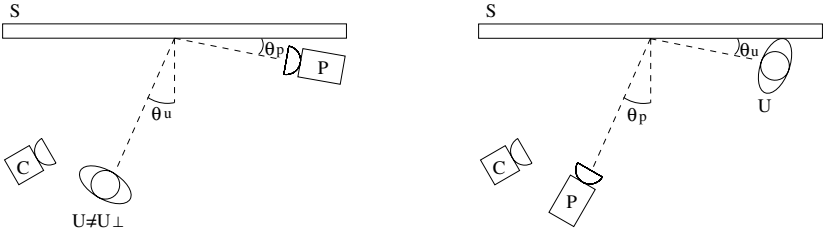


Fig. 6. *(Left)* Self-shadow avoidance. *(Right)* Virtual anamorphosis.

Fig. 7. Virtual anamorphosis. *(Left)* Frontoparallel view. *(Right)* Slanted view.

on the left the case in which the projector is slanted as much as possible with respect to the screen, so as to avoid the phenomenon of user self-shadowing and dazzling: This can be obtained without the need of expensive and cumbersome equipment such as that required for rear video projection.

On the right is reported instead the case of *virtual anamorphosis*, where the image I_p is very much deformed, and is correctly intelligible only for users observing the screen plane from a slanted viewpoint. Indeed, virtual anamorphosis is a means to convey significant visual information only to a restricted number of people around a selected viewpoint; as such, it is a way to implement a sort of *directional vision* as the analogous of directional audio. A typical operational scenario for virtual anamorphosis is when the image is projected onto the pavement or the ceiling of a room (see Fig. 7).

3.3 Arbitrary Viewpoint Change

Fig. 8 illustrates the geometry for arbitrary viewpoint change. In this scenario, the user can look at the image of a 3D object from a viewpoint that can greatly differ from the projection center used to obtain the image.

A possible application is in the architectural domain, where the photograph of the façade of a building taken from a non frontoparallel viewpoint can be interactively rectified so as to eliminate perspective distortions. Another inter-

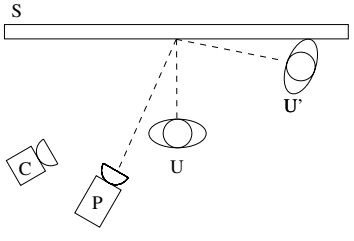


Fig. 8. Arbitrary viewpoint change: Geometry

Fig. 9. Arbitrary viewpoint change. *(Left)* Initial (frontoparallel) view. *(Middle)* Final (anamorphic) view. *(Right)* Particular of the skull.

esting application is in the software-controlled visualization of paintings. For instance, Fig. 9 shows arbitrary viewpont change applied to the painting "The ambassadors" by Hans Holbein. The new viewpoint selected corresponds to the auxiliary viewpoint chosen by the artist to hide an anamorphic enigma inside his painting. The enigma is solved by a suitable interactive warping of the original painting.

References

1. M. Ashdown, M. Flagg, R. Sukthankar, and J. Rehg, "A flexible projector-camera system for multi-planar displays." In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. II:165–172, IEEE 2004.
2. C. Colombo, A. Del Bimbo and A. Valli, "Visual capture and understanding of hand pointing actions in a 3D environment." IEEE Trans. on SMC(B), 33(4):677–686, IEEE 2003.
3. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.
4. C. Kirstein and H. Muller, "Interaction with a projection screen using a camera-tracked laser pointer." In *Proc. Int. Conf. on Multimedia Modeling (MMM)*, pp. 191-192, IEEE 1998.
5. D. Olsen Jr. and T. Nielsen, "Laser pointer interaction," in *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems*, pp. 17–22, ACM 2001.
6. R. Raskar and P. Beardsley, "A self correcting projector," In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 504–508, 2001.
7. J. M. Rehg, M. Flagg, T.-J. Cham, R. Sukthankar, and G. Sukthankar, "Projected light displays using visual feed-back." In *Proc. 7th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, pp. II:926–932, 2002.
8. R. Sukthankar, R. G. Stockton, and M. D. Mullin, "Smarter presentations: exploiting homography in camera-projector systems." In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 247–253, IEEE 2001.

Real-Time Avatar Animation Steered by Live Body Motion

Oliver Schreer[1], Ralf Tanger[1], Peter Eisert[1], Peter Kauff[1], Bernhard Kaspar[2], and Roman Englert[3]

[1] Fraunhofer Institute for Telecommunications/Heinrich-Hertz-Institut, Einsteinufer 37, 10587 Berlin, Germany
{Oliver.Schreer, Ralf.Tanger, Peter.Eisert, Peter.Kauff}@fraunhofer.hhi.de
http://ip.hhi.de
[2] T-Systems International GmbH, Am Kavalleriesand 3, 64295 Darmstadt, Germany
Bernhard.Kaspar@t-systems.com
[3] Deutsche Telekom AG, Laboratories, Ernst-Reuter-Platz 7, 10587 Berlin, Germany
roman.englert@telekom.de

Abstract. The future customer service provided by call centres will be changed due to new web-based interactive multimedia technologies. Technical support will be offered in a completely new way by using advanced image processing technologies and natural representation of virtual humans. We present a prototype system of an animated avatar, which is steered by live body motion of the operator in a call centre. The hand and head motion is transferred directly to the avatar at the customer side in order to support a more natural representation of the virtual human. The system tracks the operators hands and the head motion quite robust in real-time without specific initialization based on a monocular camera.

1 Introduction

Tracking human bodies and faces has received a lot of attention in computer vision research in the last years. The reason is, that a number of interesting applications have been raised in the past such as motion capture for entertainment industry or medical purposes, human-machine interaction, automatic surveillance systems or interactive web-based commercial applications. A lot of robust approaches have been developed, which are now going to be carried over to commercially available systems. Therefore, a lot of new challenges like robustness under different lightning conditions, independency from different users, eased use without sophisticated initialization procedures turn out. In this paper, a call centre application will be presented, where an operator is represented to the customer via an animated avatar. The head and body motion of the operator is immediately transferred to the virtual human by using robust skin colour segmentation and facial feature tracking algorithms. The complete image processing is performed on monocular colour video images in real-time on a conventional PC at full video frame rate.

Tracking of human bodies and faces as well as gesture recognition has been studied for a long time and many approaches can be found in the literature. A survey

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 147 – 154, 2005.

on human body tracking is given in [1]. A real-time body tracking system using structured light without use of additional markers is presented in [2]. This constraint is particularly important in user-friendly applications. Hand gesture recognition is reviewed in [3] and a 3D hand gesture recognition system is presented in [4]. Tracking the user's face and estimating its pose from monocular camera views is another important issue. As the 3D information is lost during perspective projection onto the image plane, some model assumptions have to be applied in order to estimate the 3D pose [5].

In [6], some specific face features are tracked in order to recover the orientation and position of the users head. Other methods use IR illumination, which simplifies tracking of the eyes [7]. In the considered scenario of animating a virtual human, the accuracy of 3D positions of head and hands does not play that important role, but the immediate transfer of general live motion to the virtual human is required such as waving hands, pointing gestures or nicking the head. This allows some simplifications in terms of accuracy, but introduces additional challenges in terms of smoothness and reliability of the animated motion. In the next section, the system of a call centre application is presented. Although this system also includes speech analysis, the focus of this paper is on image processing. Hence, the skin-colour segmentation and facial feature extraction is reported briefly. Based on the specific aims of this application, the reconstruction of the hand and head position and the head orientation is explained. Finally, results are shown and a conclusion is given.

2 System Overview

As shown in Fig. 1, the considered application provides for an operator on the sender side, who is captured by a video camera mounted on top of the display. Based on the video information, the position of the hands and the head orientation are registered and converted to standard facial and body animation parameters as standardised MPEG-4 (Part 2 (Visual)).

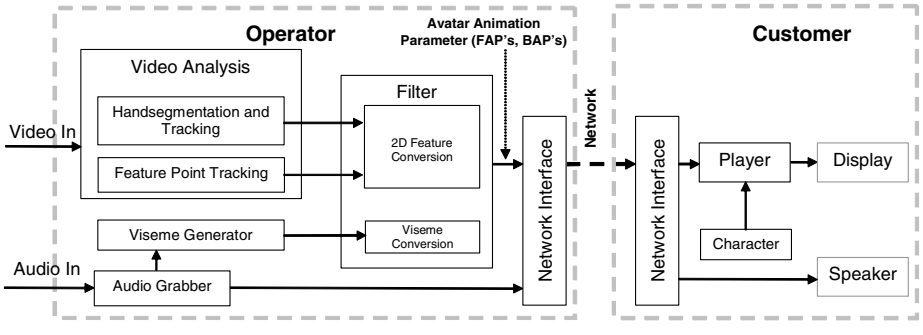


Fig. 1. Block diagram of the call centre application using an animated avatar

In addition, the voice is captured and the audio signal is transmitted to the customer at the receiving side. The captured voice is analyzed and visemes (a visual representation of phonemes) are generated to animate the lip shape corresponding to

different sounds. Based on these visemes, a natural lip movement of the avatar can be reproduced. Beside the depicted modules in Fig. 1, additional modules are implemented in order to provide more complex and natural animation. For instance, while loading predefined sequences of motion parameters from a library, the operator can activate specific high-level feature animations like opening sessions, leave-taking scenes or pointing gestures. If tracking and high-level feature animation are both switched off due to certain reasons, the natural behaviour of the animated avatar is improved by slight random motion of the head and parts of the face (e.g. eye blinking). General facial expressions like friendliness, anger or fear, which are difficult to extract from live video images can be chosen by an additional expression generator. All head and hand motions, facial expressions and lip movement are described via body animation parameters (BAP) and facial animation parameters (FAP) according to the definition in the MPEG-4 standard. The complete set of animation parameters and the audio signal are then transmitted to the customer on the receiving side. As no video information is necessary, this approach is efficient in terms of the required bandwidth and therefore appropriate in web-based customer care applications. The customer is viewing a virtual human represented by an avatar, which is steered by the live body motion and speech. In the next two sections, more details will be presented regarding the image processing part of the system. The skin-colour segmentation algorithm that is used for tracking hand and head regions is explained first. Then, the algorithm, which derives the 3D position of the hands from the corresponding segments and which is used for steering the hand movements of the avatar, is presented. Subsequently, the algorithm for tracking head features is described, and it is explained how it is used for animating head rotation. Finally, some experimental results are shown and a conclusion ends the article.

3 Skin-Colour Segmentation and Tracking

The colour of human skin is a striking feature to track and to robustly segment the operators hands and face. It is exploited, that human skin colour is independent on the human race and on the wavelength of the exposed light [8]. The same observation can be made considering the transformed colour in common video formats. Hence, the human skin-colour can be defined as a "global skin-colour cloud" in the colour space [9]. This is utilised successfully in a fast and robust region-growing based segmentation algorithm [10]: The skin colour segmentation is performed on predefined thresholds in the U,V-space of the video signal. Then, a blob recognition identifies the hands and the head in the sub-sampled image. Based on this information, a region growing approach segments the complete skin-colour region of the hands and the head quite accurately (see Fig. 2).

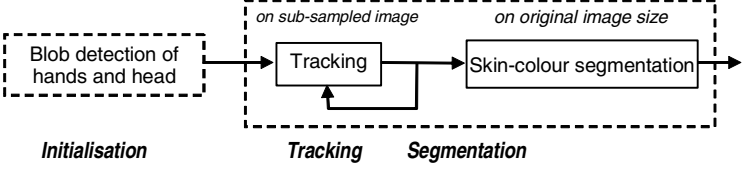


Fig. 2. Block diagram of the segmentation and tracking method of the hands

The initialisation is performed and segmentation and tracking start as soon as three separated skin-colour blobs are detected. The blobs will be assigned to hands and the head supposing that hands are initially below the head, which holds for general poses. The approach achieves real-time performance due to tracking on sub-sampled images and skin-colour segmentation limited to bounding boxes circumscribing the hands and the head. Results are presented in section 6, Fig. 7 and Fig. 6.

In this context, it is a specific problem to resolve overlapping between different skin-coloured regions such as hands and the head. If a hand has contact with the face, the following process is carried out: In addition to the hands, the head blob of the participant resulting from the initialisation phase is tracked as well in the sub-sampled image, using a third bounding box. If one of the hand boxes overlaps with the head box, then only the non-overlapping part of the hand box is considered for tracking the centre of gravity. More details can be found in [10].

4 Facial Feature Extraction

The aim of facial feature tracking is to obtain sufficient information in order to derive a convincing and reliable rotation of the operator's head. As a result from the segmentation and tracking algorithm described in the previous section, the bounding box of the operator's head is used as a starting point for facial feature extraction. The skin-coloured pixels inside the bounding box are marked and a standard feature tracker is applied to this limited face region. The feature tracker is based on a two-step approach. First, relevant features are selected by using corner operators such as Moravec or Harris detectors. Secondly, the selected features are then tracked continuously from frame to frame by using a feature dissimilarity measure. This guarantees, that features are discarded from further tracking in the case of occlusions. Even in the case of a rotating head some good features become distorted due to perspective changes or even become invisible and get lost. In Fig. 3, markers of selected features are shown in the face region in three succeeding frames. The big cross assigns the median value of all skin coloured pixels. The considered skin colour region is marked by the line around the face. Due to the blond hairs of the test person, the hairs are recognized as well as skin.

Fig. 3. Facial feature tracking result of three succeeding frames

5 Reconstruction of Head Orientation and Hand Positions

The main goal of the application from section 2 is to animate an avatar by human body motion captured from real-time video. Hence, accuracy in terms of correct 3D positions of the hands or precise nick and turn angles of the head are not required. However, reliable, convincing and smooth motions are important in order to support natural representation of a dynamic virtual human. On one hand, this fact facilitates the estimation of animation parameters in some way and can be exploited for simplifications. On other hand, the extracted parameters have to be filtered and outliers must be discarded in order to provide smooth motion.

The head orientation is derived from the results provided by a facial feature tracker. Based on a few robustly tracked facial features, the head orientation can be analysed by comparing the relative motion of facial features to the projected 2D motion of the head. This 2D motion is calculated by the mean change of position of all face pixels in succeeding frames. The task is to distinguish between head rotation and pure translation. In the case of a pure translation, the relative motion of each feature compared to the motion of the mean of all face pixel positions should be zero. Just the opposite holds in the case of the rotation. In this case, the motion of the mean of all face pixel positions should be significantly smaller than the relative motion of the facial features. This behaviour of facial feature points allows a simple approximation of the head rotation in horizontal (turn angle) and vertical direction (nick angle). The median value of horizontal and vertical coordinates of facial feature points is assigned with $(\overline{m}_t, \overline{n}_t)$, whereas the mean of all face pixel positions is denoted by $(\overline{p}_t, \overline{q}_t)$. The relative change of facial feature points (horizontal/vertical) is then calculated by Equ. 1 and the change of horizontal and vertical rotation is approximated by Equ.2. A scale factor γ is introduced to adopt the pixel unit to angle.

$$\Delta u = (\overline{m}_t - \overline{m}_{t-1}) - (\overline{p}_t - \overline{p}_{t-1}), \qquad \Delta v = (\overline{n}_t - \overline{n}_{t-1}) - (\overline{q}_t - \overline{q}_{t-1}). \tag{1}$$

$$\Delta\varphi_u = \sin\left(\gamma \cdot \Delta u \cdot \tfrac{\pi}{180}\right), \qquad \Delta\varphi_v = \sin\left(\gamma \cdot \Delta v \cdot \tfrac{\pi}{180}\right). \tag{2}$$

As it is obviously not possible to calculate the absolute rotation from this method, drift effects may occur. This can be avoided by continuously weighting the current turn (or nick) angle by some factor less than 1. As the central viewing direction is the most relevant, the animated head will adopt to this position after a while.

The positions of the hands are derived from the result of the skin-colour segmentation and tracking module. It provides reasonable and stable results of the motion of both hands in the 2D image plane. To achieve natural avatar movements, the 2D positions have to be transferred onto the 3D model of the avatar. Since two degrees of freedom of the hand position are available, just a simplified motion model can be implemented in this case. Therefore the system is based on the assumption, that the hands of the animated avatar mainly move within a 2D plane in the 3D space. Thus, taking into account some further physical constraints such as the restricted range of elbow joint and the proportions between the upper arm and the forearm, the position of the avatar's hands can be computed from these 2D tracking results. Nevertheless, a 2D to 3D reprojection is necessary, which requires some knowledge about the imaging process. The general projection equation for a 3D point M_w in world coordinates into a 2D point **m** in image coordinates is as follows:

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{M}_w \quad \text{with} \quad \mathbf{P} = \begin{bmatrix} \mathbf{p}_1^T & p_{14} \\ \mathbf{p}_2^T & p_{24} \\ \mathbf{p}_3^T & p_{34} \end{bmatrix} = \begin{bmatrix} a_x\mathbf{r}_1^T + u_0\mathbf{r}_3^T & a_x t_x + u_0 t_z \\ a_y\mathbf{r}_2^T + v_0\mathbf{r}_3^T & a_y t_y + v_0 t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \tag{3}$$

The matrix **P** is called the general projection matrix containing the internal camera parameters (u_0, v_0, a_u, a_v) and the external parameters (**R**, **t**=(t_x, t_y, t_z)) relating the world coordinate system to the camera coordinate system. For both components of the 2D image point, we get the following two reconstruction equations:

$$(1)\quad \left(\mathbf{p}_1 - u\,\mathbf{p}_3\right)^T M_w + p_{14} - u\,p_{34} = 0, \qquad (2)\quad \left(\mathbf{p}_2 - v\,\mathbf{p}_3\right)^T M_w + p_{24} - v\,p_{34} = 0 \tag{4}$$

These two equations have in general three unknowns, the three components of the 3D point, which can only be solved, if a second image of another camera is available, the so-called stereo case. As mentioned previously, it is assumed, that the hand position is fixed in a predetermined depth plane, which results in a fixed and known Z_w coordinate. In this case, a reconstruction becomes possible. Furthermore, we are able to avoid precise calibration as just a general transformation between the 2D image plane and the 3D plane at fixed depth is required. The setup of the camera related to the world coordinate system is shown in Fig. 4. If we assume just a horizontal rotation of the camera, a translational shift in y- and z-direction and a coinciding origin of the image coordinate system with the principal point, we get the following simplified projection matrix (see Equ. 5). The reconstruction equation (Equ. 4) for the desired X_w and Y_w coordinates becomes quite simple as shown in Equ. 6.

$$\mathbf{P}' = \begin{bmatrix} a_x & 0 & 0 & 0 \\ 0 & a_y r_{22} & a_y r_{23} & a_y t_y \\ 0 & r_{32} & r_{33} & t_z \end{bmatrix}, \quad \text{with} \quad \mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r_{22} & r_{23} \\ 0 & r_{32} & r_{33} \end{bmatrix}, \quad u_0 = v_0 = t_x = 0 \tag{5}$$

$$\begin{aligned} &a_1 X_W + a_2 Y_W + a_3 = 0 \quad \text{with} \quad a_1 = a_u, \quad a_2 = -u\,a_v\,r_{23}, \quad a_3 = -\left(u\,r_{33}\,Z_w + u\,t_z\right) \\ &b_1 Y_W + b_2 = 0 \quad \text{with} \quad b_1 = a_v\,r_{22} - v\,a_v\,r_{23}, \quad b_2 = \left(r_{32} - v\,r_{33}\right)Z_w + a_v\,t_y - v\,t_z \end{aligned} \tag{6}$$

The resulting reconstructed point in world coordinates is finally:

$$X_W = a_2 b_2 - b_1 a_3 / a_1 b_1, \qquad Y_W = -b_2 / b_1 \tag{7}$$

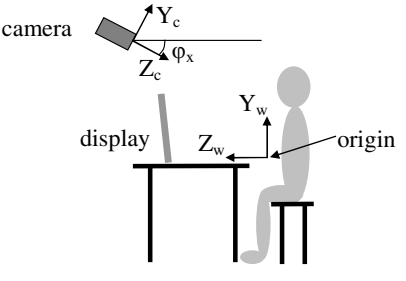


Fig. 4. Simplified camera setup of the considered scenario

The horizontal rotation and the translation of the camera have been measured approximately, whereas the horizontal and vertical scale factor have been chosen experimentally.

6 Prototype Results

The presented approach is fully integrated in a real-time application including transmission of animation parameters and display of the animated avatar. In the following figures, several snap shots of a video sequence are presented, which show the actual pose of the user and the animated pose of the virtual character. In the right part of each example, the box around the head and the hand boxes including the segmented skin-colour pixels are shown. In Fig. 5 and Fig. 6 (right), the transformed head rotation is demonstrated resulting from the 2D video images of the operator. In Fig. 6 (left), a sequence of head nick and turn angles is presented. The sinusoidal behaviour resulting from up and down (left and right) motion, is clearly visible. Interestingly, the nick angle changes since horizontal head turn was performed. The reason is caused by the mounting of the camera, which is looking from 0.4m above the head with an angle of 30 degrees. In Fig. 7, the animation of the avatar by moving hands is shown.

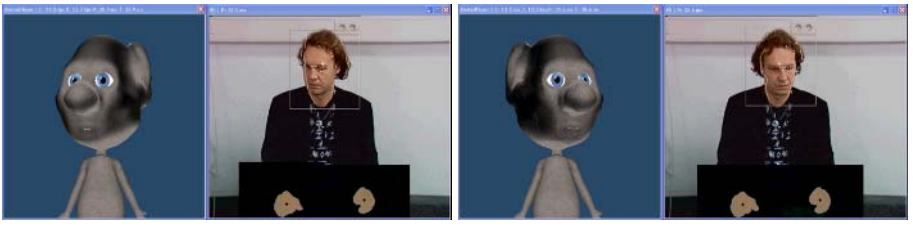


Fig. 5. Example snap shots for a head turn

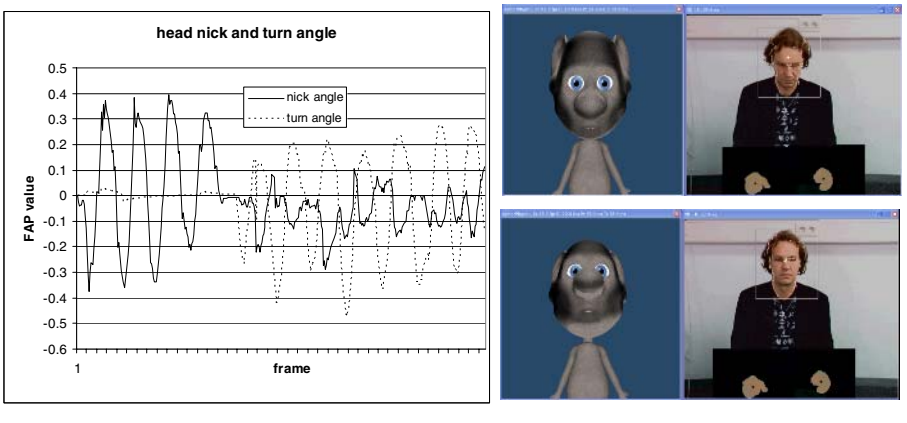


Fig. 6. Sequence of registered nick and turn angles described as FAP value (left), example images for nicking the head (right)

Fig. 7. Example snap shots for moving hands

7 Conclusion

In this paper, we have presented a complete system which uses several modules in order to steer an avatar based on live motion of a person captured by a single camera. The approach is running in real-time on a standard PC. The algorithms are robust in terms of different users, arbitrary gestures and with regard to the initialisation of the complete tracking and segmentation system. An automatic initialisation prevents the user from difficult setup procedures or specific initial gestures. This is particularly important in consumer applications, where user friendliness and easy usage play a significant role.

References

1. T.B. Moeslund and E. Granum (2001) A survey of computer vision-based human motion capture, Computer Vision and Image Understanding, vol. 81, no. 3, 231-268.
2. T. Jaeggli, T.P. Koninckx and L. Van Gool (2005) Model-based Sparse 3D Reconstructions for Online Body Tracking. Proceedings of IS&T/SPIE's 17th Annual Symposium on Electronic Imaging - Videometrics VIII, vol.5665, San Jose, California, USA.
3. I. Pavlovic, R. Sharma, T.S. Huang (1997) Visual interpretation of hand gestures for human–computer interaction: a review. IEEE Trans. on PAMI, 19: 677–695.
4. A. Just, S. Marcel and O. Bernier (2004) HMM and IOHMM for the recognition of Mono- and Bi-manual 3D Hand Gestures. British Machine Vision Conf., Kingston Univ. London.
5. P. Eisert and B. Girod (1998) Analyzing Facial Expressions for Virtual Conferencing. IEEE Computer Graphics & Applications: Special Issue: Computer Animation for Virtual Humans, vol. 18, no. 5, pp. 70-78.
6. T. Horprasert, Y. Yacoob, L.S. Davis (1996) Computing 3-D head orientation from a monocular image sequence. 2nd Int. Conf. on Automatic Face and Gesture Recogn. , p.242.
7. Z. Zhu, Q. Ji (2004) 3D Face Pose Tracking from an Uncalibrated Monocular Camera. Int. Conf. on Pattern Recognition, Workshop on Face Processing in Video, Washington DC.
8. R. R. Anderson, J. Hu, and J. A. Parrish (1981) Optical radiation transfer in the human skin and applications in in vivo remittance spectroscopy. In R. Marks and P. A. Payne, editors, Bioengineering and the Skin, MTP Press Limited, chapt. 28, pp. 253-265.
9. M. Störring, H.J. Andersen, E. Granum, (1999) Skin colour detection under changing lighting conditions. Symp. on Intelligent Robotics Systems, pp. 187-195.
10. S. Askar, Y. Kondratyuk, K. Elazouzi, P. Kauff, O. Schreer (2004) Vision-based Skin-Colour Segmentation of Moving Hands for Real-Time Applications. Proc. of 1st European Conference on Visual Media Production (CVMP), London, United Kingdom.

Vision-Based Registration for Augmented Reality with Integration of Arbitrary Multiple Planes

Yuko Uematsu and Hideo Saito

Keio University, Dept. of Information and Computer Science, Yokohama, Japan
{yu-ko, saito}@ozawa.ics.keio.ac.jp
http://www.ozawa.ics.keio.ac.jp/Saito

Abstract. We propose a novel vision-based registration approach for Augmented Reality with integration of arbitrary multiple planes. In our approach, we estimate the camera rotation and translation by an uncalibrated image sequence which includes arbitrary multiple planes. Since the geometrical relationship of those planes is unknown, for integration of them, we assign 3D coordinate system for each plane independently and construct projective 3D space defined by projective geometry of two reference images. By integration with the projective space, we can use arbitrary multiple planes, and achieve high-accurate registration for every position in the input images.

1 Introduction

Augmented Reality (AR) / Mixed Reality (MR) is a technique which can superimpose virtual objects onto the real 3D world. We can see the virtual objects as if they really exist in the real world, so AR provide users with more effective view [1,2]. One of the most important issues for AR is geometrical registration between the real world and the virtual world. In order to achieve correct registration, accurate measurements of the camera rotations and translations (corresponding to the user's view) are required.

For the measurements, some kind of sensors such as magnetic or gyro sensors may be used. The registration by such sensors is stable against a change in light conditions and is especially effective when a camera moves rapidly. However, the rotations and translations obtained from sensors are not accurate enough to achieve perfect geometrical registration. Furthermore, the use of sensors has some limitations in practice: user's movable area, perturbation caused by the environment, and so on. On the other hand, vision-based registration does not require any special devices expect cameras. Therefore an AR system can be constructed easily. This kind of registration relies on the identification of features in the input images. Typically artificial markers placed in the real world [3,4], prepared model [5,6,7], and / or natural features are used for the registration. Related works based on natural features have used various features: feature points [8], edges or curves. However, it is also true that few features are available for registration in the real world. Therefore, it is important how to use the few features effectively.

We focus on the planar structures, which exist in the real world naturally without artificial arrangement and put appropriate restrictions to the natural feature points. Since

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 155–162, 2005.

a lot of planes exist in various environments, such as indoor walls, floors, or outdoor wall surfaces of buildings etc., using these structures is very reasonable approach. Using multiple planes, we can overlay virtual objects onto wider area than using only 1 plane and the accuracy is also improved. Furthermore, using multiple planes which are in arbitrary positions and poses, we can use most planes existing in the real world. Therefore, using arbitrary multiple planes is valuable approach for the future AR applications.

Registration using planes has attracted attention recently, and Simon et al. have proposed related AR approaches [9,10,11]. In [9], they track feature points existing on a plane in the real world, estimate a projection matrix for each frame by the tracked points and overlay virtual objects onto the real input images. They also implement registration using multiple planes. In [10], they estimated the projection matrix by multiple planes which are perpendicular to the reference plane using an uncalibrated camera. In [11], they estimated the projection matrix using a calibrated camera from multiple planes of arbitrary positions and poses. In their method, the geometrical relationship among these planes and motion of the camera are calculated by bundle adjustment which is carried out over all frames.

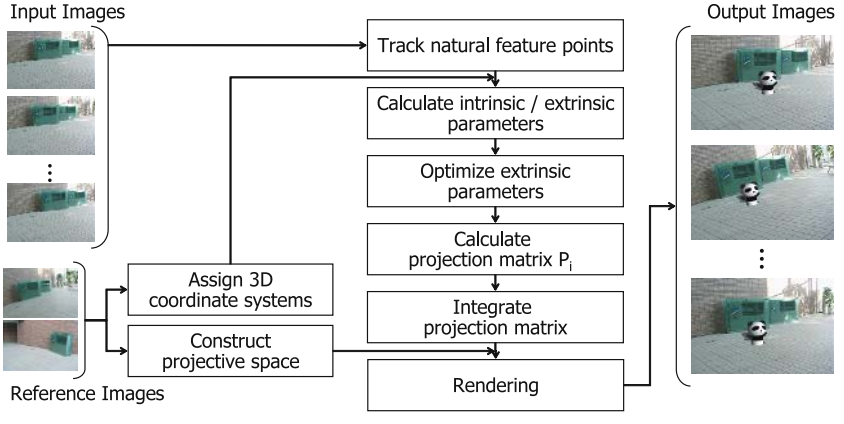


Fig. 1. Overview of the proposed method

In this paper, we propose a vision-based registration approach, which can use arbitrary multiple planes without any information on physical relationship of the planes, estimate the camera motion frame by frame, and achieve high registration accuracy. In order to use arbitrary multiple planes, we assign 3D coordinate systems for each plane independently. Since geometrical relationship among those planes is unknown, we construct "projective 3D space" for integrating those planes. This integration is main contribution of our approach. Fig.1 describes an overview of our approach. Firstly, the input image sequence in which n planes exist is taken by an uncalibrated hand-held video camera. Next, we compute each projection matrix (corresponding to the camera motion) from each plane independently. Then, the projective space is constructed by 2 reference images, which are taken at 2 views, those matrices are integrated with the space, so one camera motion is obtained. Lastly, virtual objects are overlaid onto the input images according to the camera motion.

2 Calculation of Projection Matrix

As mentioned previously, for overlaying virtual objects, accurate measurement of the camera rotations and translations (extrinsic parameters of the camera) is required. Moreover, for using an uncalibrated camera, we also need to estimate intrinsic parameters. In our approach, we assign a 3D coordinate system for each plane so that each plane is set to $Z = 0$ (see sec.2.1). Then we compute intrinsic and extrinsic parameters from a homography between the 3D real plane and the input image plane in order to obtain a projection matrix [9].

A 3D coordinate system is related to a 2D coordinate system by 3×4 projection matrix $\boldsymbol{P}$. Thus, each 3D coordinate system designed for each plane is also related to the input images by each projection matrix. If each plane's Z coordinate is set to 0, a homography $\boldsymbol{H}$ also relates between each plane and the input images.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \simeq \boldsymbol{P} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} \simeq \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \simeq \hat{\boldsymbol{P}} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \simeq \boldsymbol{H} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (1)$$

This 3×3 matrix (called $\hat{\boldsymbol{P}}$), which is the deleted third column of $\boldsymbol{P}$, is equivalent to a planar homography $\boldsymbol{H}$. The deleted column vector can be estimated by this $\boldsymbol{H}$. When the homography is calculated, we can obtain the projection matrix from it. In particular, we think dividing into intrinsic parameters $\boldsymbol{A}$, and extrinsic parameters $\boldsymbol{R}$, $\boldsymbol{t}$.

$$\boldsymbol{P} = \boldsymbol{A}\,[\boldsymbol{R} \mid \boldsymbol{t}] = \boldsymbol{A}\,[\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{r_3}\ \boldsymbol{t}] \quad (2)$$

$$\hat{\boldsymbol{P}} = \boldsymbol{A}\,[\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{t}] = \boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (3)$$

2.1 Assigning of 3D Coordinate Systems

For using the multiple planes whose geometrical relationship is unknown, we assign a 3D coordinate system for each plane in the 3D real world independently. Each plane's Z coordinate is set to 0. This is for computing a homography and estimate a projection matrix from it.

2.2 Calculation of Homography

In order to estimate the intrinsic and the extrinsic parameters, we calculate homographies between each 3D plane ($Z = 0$) and the input image plane. Natural feature points existing on the 3D planes are tracked by KLT-feature-tracker [12] and used for computing the homography. The Homography is calculated for each 3D plane independently, so homographies and projection matrices are computed to the number of the 3D planes respectively.

2.3 Estimation of Intrinsic Parameters

By fixing the skew to 0, the aspect ratio to 1 and the principal point to the center of the image, the intrinsic parameters can be defined as in eq.(4), and the relationship to homography is represented by eq.(5). Then, we only have to estimate the focal length f.

$$\boldsymbol{A} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{matrix} (c_x, c_y) : \text{ principal point} \\ f \quad : \text{ focal length} \end{matrix} \tag{4}$$

$$\boldsymbol{A^{-1}H} = [\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{t}] \tag{5}$$

According to the property of rotation matrix $\boldsymbol{R}$, that is the inner product of $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ is equal to 0, we can calculate the focal length f.

$$f^2 = \frac{(h_{11} - c_x h_{31})(h_{12} - c_x h_{32}) + (h_{21} - c_y h_{31})(h_{22} - c_x h_{32})}{-h_{31} h_{32}} \tag{6}$$

2.4 Estimation of Extrinsic Parameters

The extrinsic parameters of a camera consist of a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$. Since $\boldsymbol{r_1}$, $\boldsymbol{r_2}$ (the first and second column vectors of $\boldsymbol{R}$) and $\boldsymbol{t}$ are already known, we should estimate only $\boldsymbol{r_3}$. Then, also according to the property of $\boldsymbol{R}$, that is the cross product of $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ becomes $\boldsymbol{r_3}$, we compute $\boldsymbol{r_3}$. Therefore, $\boldsymbol{R}$ is

$$\boldsymbol{R} = [\boldsymbol{r_1}\ \boldsymbol{r_2}\ (\boldsymbol{r_1} \times \boldsymbol{r_2})] = [\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{r_3}] \tag{7}$$

Furthermore, the extrinsic parameters are optimized by the steepest descent method in order to improve its accuracy. We optimize errors ϵ between the initial point $\boldsymbol{x_p}$ projected by the calculated projection matrix and the point $\boldsymbol{x_h}$ by homography.

$$\epsilon = (\boldsymbol{x_h} - \boldsymbol{x_p}) \tag{8}$$

3 Integration of Projection Matrices

Our main contribution is using multiple planes whose geometrical relationship is unknown and existing in the real world arbitrarily. After assigning 3D coordinate system for each plane independently and calculating projection matrices, we integrate those projection matrices in order to use the information of multiple planes. Each projection matrix is reliable around its corresponding plane, however, as the position of a virtual object moves away from each plane, the accuracy becomes lower. Therefore, we integrate the projection matrices to compute one accurate matrix over the whole image. However, it is impossible to integrate them simply because each projection matrix is from different 3D coordinate system. Then, we construct projective 3D space based on the projective geometry of two reference images.

If there are n planes, the relationship among each 3D coordinate system assigned for each plane, the projective space, and the input images is shown in fig.2. Since this

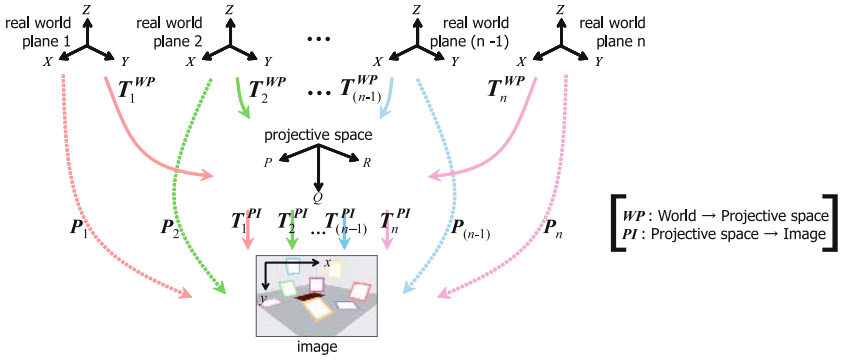


Fig. 2. Relationship among 3 coordinate sysmtes

projective space is defined by only projective geometry of two reference images, it is independent from the 3D coordinate system of the planes. Thus, $\boldsymbol{T}_{\boldsymbol{k}}^{\boldsymbol{PI}}$ are the transformation matrices between the common coordinate sysmtes (projective space → input images), and we can integrate those matrices. In this way, using projective space, we can extract the common transformation matrices from the porjection matrices calculated from different 3D coordinate sysmtems and integrate arbitrary multiple planes. This integration of multiple planes, which are different poses and exist in various positions, allows accurate augmentation onto wider area than using only 1 plane. The detail will be described below.

3.1 Construction of Projective Space

The projective space used in our method is based on the concept of "projective reconstruction" as shown in fig.3. By epipolar geometry between the reference images (cameras), the relationship between the projective space and the reference images is as follows respectively,

$$\boldsymbol{P}_{\boldsymbol{A}} = [\mathbf{I} \mid \mathbf{0}], \qquad \boldsymbol{P}_{\boldsymbol{B}} = [\boldsymbol{M}\boldsymbol{e}_{\boldsymbol{B}}], \qquad \boldsymbol{M} = -\frac{[\boldsymbol{e}_{\boldsymbol{B}}]_{\times}\boldsymbol{F}_{\boldsymbol{AB}}}{\|\boldsymbol{e}_{\boldsymbol{B}}\|^2} \tag{9}$$

where $\boldsymbol{F}_{\boldsymbol{AB}}$ is a fundamental matrix of image A to B, and $\boldsymbol{e}_{\boldsymbol{B}}$ is an epipole on the image B. Consider $\boldsymbol{C}_{\boldsymbol{p}}$ as a point in the projective space, $\boldsymbol{C}_{\boldsymbol{A}}(u_A, v_A)$ as on the image A, $\boldsymbol{C}_{\boldsymbol{B}}(u_B, v_B)$ as on the image B, we can write

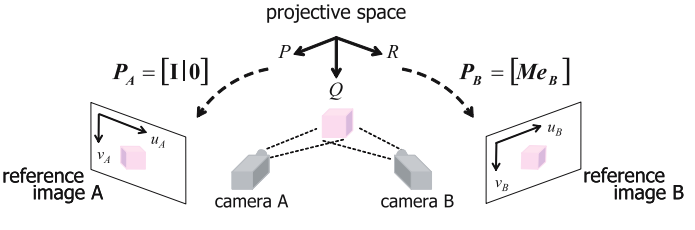


Fig. 3. Projective space by projective reconstruction

$$\boldsymbol{K}\boldsymbol{C}_{\boldsymbol{P}} = \left[\ \boldsymbol{p}^{\boldsymbol{1}}_{\boldsymbol{A}} - u_{A}\boldsymbol{p}^{\boldsymbol{3}}_{\boldsymbol{A}},\ \boldsymbol{p}^{\boldsymbol{2}}_{\boldsymbol{A}} - v_{A}\boldsymbol{p}^{\boldsymbol{3}}_{\boldsymbol{A}},\ \boldsymbol{p}^{\boldsymbol{1}}_{\boldsymbol{B}} - u_{B}\boldsymbol{p}^{\boldsymbol{3}}_{\boldsymbol{B}},\ \boldsymbol{p}^{\boldsymbol{2}}_{\boldsymbol{B}} - v_{B}\boldsymbol{p}^{\boldsymbol{3}}_{\boldsymbol{B}}\ \right]^{\top}\boldsymbol{C}_{\boldsymbol{P}} = \boldsymbol{0} \quad (10)$$

$\boldsymbol{p}^{i}$ is the i th column vector of $\boldsymbol{P}$. Then, we obtain $\boldsymbol{C}_{p} \simeq [p, q, r, 1]^{\top}$ by the singular value decomposition of $\boldsymbol{K}$.

3.2 Calculation of $\boldsymbol{T}^{\boldsymbol{WP}}_{\boldsymbol{k}}$

Consider $\boldsymbol{C}_{\boldsymbol{W}}(X, Y, Z)$ as a point on the kth plane in the real world and $\boldsymbol{C}_{\boldsymbol{P}}(P, Q, R)$ as in the projective space, the relationship between the two coordinate systems is

$$\boldsymbol{C}_{\boldsymbol{P}} \simeq \boldsymbol{T}^{\boldsymbol{WP}}_{\boldsymbol{k}}\boldsymbol{C}_{\boldsymbol{W}} \quad (11)$$

Since $\boldsymbol{T}^{\boldsymbol{WP}}_{\boldsymbol{k}}$ is $4{\times}4$ matrix, we can compute this matrix by the 5 (or more) corresponding points, in which any combination of 3 points must not be colinear and 4 points must not also be coplanar.

3.3 Calculation of $\boldsymbol{T}^{\boldsymbol{PI}}_{\boldsymbol{k}}$

When $\boldsymbol{T}^{\boldsymbol{WP}}_{\boldsymbol{k}}$ is known, we can compute $\boldsymbol{T}^{\boldsymbol{PI}}_{\boldsymbol{k}}$ by eq.(12), so $\boldsymbol{T}^{\boldsymbol{PI}}_{\boldsymbol{1}} \sim \boldsymbol{T}^{\boldsymbol{PI}}_{\boldsymbol{n}}$ are computed for each plane as fig.2.

$$\boldsymbol{T}^{\boldsymbol{PI}}_{\boldsymbol{k}} = \boldsymbol{P}_{k}(\boldsymbol{T}^{\boldsymbol{WP}}_{\boldsymbol{k}})^{-\boldsymbol{1}} \quad (12)$$

As described previously, these matrices represent the common transformation (projective space $\rightarrow$ input images). Therefore, we can integrate these matrices. For the integration, we propose two approaches.

Maximum likelihood estimation. Using $\boldsymbol{T}^{\boldsymbol{PI}}_{\boldsymbol{k}}$, set of corresponding points between the projective space and the input images can be calculated for each plane. Then, we calculate $\boldsymbol{T}^{\boldsymbol{PI}}$ by the maximum likelihood estimation method using those points. This means that, if n planes exist and m set of corresponding points are calculated every plane, the integration expression becomes as follows.

$$\begin{bmatrix} X_{11} & Y_{11} & Z_{11} & 1 & 0 & 0 & 0 & 0 & -X_{11}x_{11} & -Y_{11}x_{11} & -Z_{11}x_{11} \\ 0 & 0 & 0 & 0 & 1 & X_{11} & Y_{11} & Z_{11} & -X_{11}y_{11} & -Y_{11}y_{11} & -Z_{11}y_{11} \\ & & & & & & \vdots & & & & \\ X_{nm} & Y_{nm} & Z_{nm} & 1 & 0 & 0 & 0 & 0 & -X_{nm}x_{nm} & -Y_{nm}x_{nm} & -Z_{nm}x_{nm} \\ 0 & 0 & 0 & 0 & 1 & X_{nm} & Y_{nm} & Z_{nm} & -X_{nm}y_{nm} & -Y_{nm}y_{nm} & -Z_{nm}y_{nm} \end{bmatrix} \begin{bmatrix} t^{PI}_{11} \\ t^{PI}_{12} \\ \vdots \\ t^{PI}_{33} \\ t^{PI}_{34} \end{bmatrix} = \begin{bmatrix} x_{11} \\ y_{11} \\ \vdots \\ x_{nm} \\ y_{nm} \end{bmatrix} \quad (13)$$

Merging with weights. In order to integrate $\boldsymbol{T}^{\boldsymbol{PI}}_{\boldsymbol{k}}$ in consideration for that each projection matrix is reliable around each plane, we employ the following integration form.

$$\boldsymbol{T}^{\boldsymbol{PI}} = \frac{1}{n}\left[\ w_{1}\ \cdots\ w_{n}\ \right]\left[\ \boldsymbol{T}^{\boldsymbol{PI}}_{\boldsymbol{1}},\ \ldots,\ \boldsymbol{T}^{\boldsymbol{PI}}_{\boldsymbol{n}}\ \right]^{\top} \quad (14)$$

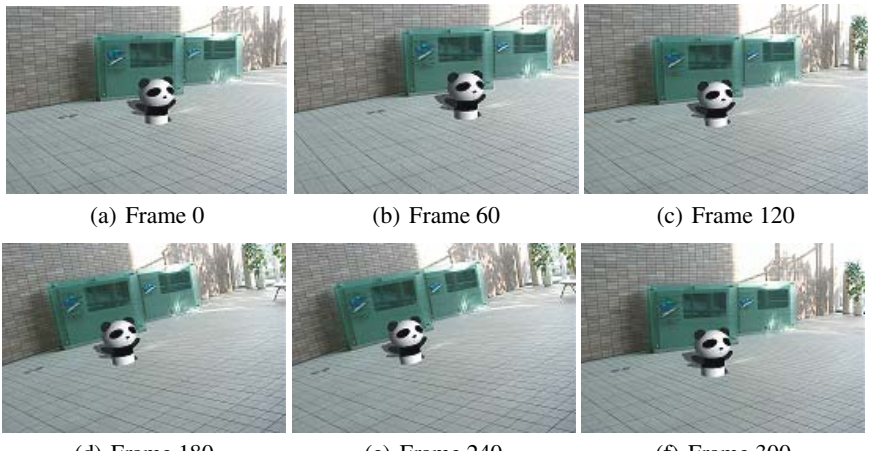
w_k is a weight parameter which is defined according to the distance from each plane to the overlaid position. This integration enables effective augmentation depending on the overlaid position. We use this one for the experiments in the next section (sec.4).

4 Experimetal Results

In this section, the experimental results are shown to prove the availability of the proposed method. We implement the AR system based on our method using only a PC (OS:Windows XP, CPU:Intel Pentium Ⅳ 3.20GHz) and a CCD camera (SONY DCR-TRV900). The input image's resolution in all the experiments is 720×480 pixels, and graphical views of a virtual object are rendered using OpenGL.

The overlaid result images produced by the augmentation are shown in fig.4. In this sequence, the 3 planes (a floor, a front display, and a back wall) are used for registration and a virtual object (a figure of panda) is overlaid on the floor plane. As shown in the figure, our approach can superimpose a virtual object onto the input images successfully.

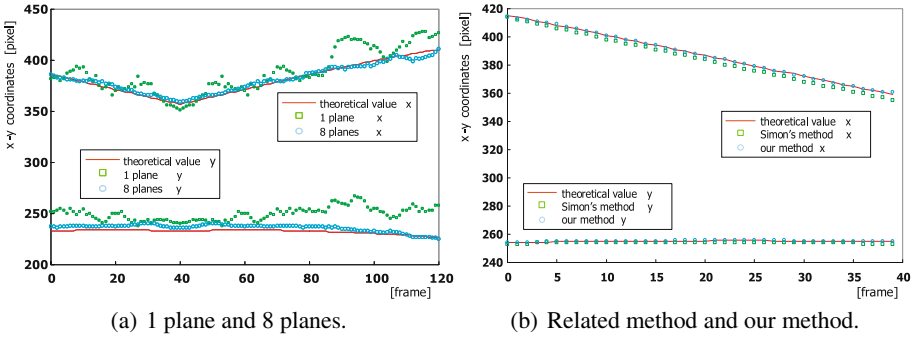
Next, in order to evaluate the registration accuracy in our method, we perform the same implementation for the synthesized images rendered with OpenGL. Since we have



(a) Frame 0 (b) Frame 60 (c) Frame 120

(d) Frame 180 (e) Frame 240 (f) Frame 300

Fig. 4. Overlaid image sequence of a virtual object



(a) 1 plane and 8 planes. (b) Related method and our method.

Fig. 5. Comparison of x-y coordinates accuracy with theoretical value

to know the exact position and pose of a camera to evaluate accuracy, we use the synthesized images. Fig.5(a) shows that the result by 8 planes has less registration errors and jitters than using only 1 plane, in spite of no information about the relationship of the planes. This suggests that increasing the number of planar structures in the scene using the proposed method can improve the registration accuracy.

We also evaluate the proposed method by comparing with one of related works by Simon [10], in which multiple planes need to be perpendicular to the reference plane (that is one of multiple planes). For the comparison, we apply the images, in which 3 orthogonal planes exist, to Simon's method and our method, and evaluate the registration accuracy. The result of the evaluation is shown in fig.5(b). Even though our method does not require any geometrical information of the plane, our method achieves almost the same accuracy with their method, in which the planes need to be perpendicular to the reference plane.

5 Conclusion

A geometrical registration method for Augmented Reality with an uncalibrated camera based on multiple planes has been proposed in this paper. The planes do not need to be perpendicular to each other. This means that any planes at arbitrary positions and poses can be used for registration. Furthermore the registration can be performed frame by frame without using all frames in the input image sequence. Thus we can construct the AR system easily, and overlay virtual objects onto the image sequence correctly.

References

1. Azuma, R.T.: A survey of augmented reality. Presence (1997) 355–385
2. Azuma, R.T.: Recent advances in augmented reality. IEEE Computer Graphics and Applications 21 (2001) 34–47
3. Billinghurst, M., et al.: Magic book: Exploring transitions in collaborative ar interfaces. Proc. of SIGGRAPH 2000 (2000) 87
4. Satoh, K., et al.: Robust vision-based registration utilizing bird's-eye with user's view. In: Proc. of the ISMAR. (2003) 46–55
5. Drummond, T., Cipolla, R.: Real-time tracking of complex structures with on-line camera calibration. In: Proc. of the BMVC. (1999) 574–583
6. Comport, A.I., Marchand, E. Chaumette, F.: A real-time tracker for markerless augmented reality. In: Proc. of the ISMAR. (2003) 36–45
7. Klein, K., Drummond, T.: Sensor fusion and occlusion refinement for tablet-based ar. In: Proc. of the ISMAR. (2004) 38–47
8. Chia, K.W., Cheok, A., Prince, S.J.D.: Online 6 dof augmented reality registration from natural features. In: Proc. of the ISMAR. (2002) 305–313
9. Simon, G., Fitzgibbon, A., Zisserman, A.: Markerless tracking using planar structures in the scene. In: Proc. of the ISAR. (2000) 120–128
10. Simon, G., Berger, M.: Reconstructing while registering: a novel approach for markerless augmented reality. In: Proc. of the ISMAR. (2002) 285–294
11. Simon, G., Berger, M.O.: Real time registration known or recovered multi-planar structures: application to ar. In: Proc. of the BMVC. (2002) 567–576
12. Shi, J., Tomasi, C.: Good features to track. IEEE Conf. on CVPR (1994) 593–600

A Kalman Filter Based Background Updating Algorithm Robust to Sharp Illumination Changes

Stefano Messelodi[2], Carla Maria Modena[2], Nicola Segata[1], and Michele Zanin[2]

[1] University of Trento, Italy,
[2] ITC-irst, Via Sommarive, 18 - 38050 Povo (Trento), Italy

Abstract. A novel algorithm, based on Kalman filtering is presented for updating the background image within video sequences. Unlike existing implementations of the Kalman filter for this task, our algorithm is able to deal with both gradual and sudden global illumination changes. The basic idea is to measure global illumination change and to use it as an external control of the filter. This allows the system to better fit the assumptions about the process to be modeled. Moreover, we propose methods to estimate measurement noise variance and to deal with the problem of saturated pixels, to improve the accuracy and robustness of the algorithm. The algorithm has been successfully tested in a traffic surveillance task by comparing it to a background updating algorithm, based on Kalman filtering, taken from literature.

1 Introduction

The most popular techniques for the detection of moving objects in complex environments observed by a static camera, are based on the background differencing method. It consists of maintaining a background image of the scene and detecting foreground objects by subtracting it from the current frame. Background updating is a critical task in outdoor scenes which undergo significant changes caused both by natural events, e.g. the sun suddenly disappearing behind clouds, and artificial events, like the change of the exposure time of the acquisition device, or the switching-on of artificial lights. A background updating module should be able to detect static background pixels, dynamic background pixels and to deal with gradual and sharp illumination changes. Many different methods have been proposed in literature that vary in their adopted features. A common classification distinguishes the techniques depending on the features they use: a) pixel level [8,2,4,3,9,10]: only the temporal distribution of intensities is used, b) region level [12]: a small neighbourhood of each pixel is considered to take into account local structure; c) spatial-temporal level [13,11]: both spatial and temporal features are used mainly to detect non-static background pixels (moving leaves, rippling water,...). Several techniques take into account the result of foreground object detection as a feedback in order to apply different updating criteria to the background and foreground regions. As background updating is

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 163–170, 2005.

a preliminary step within more complex surveillance systems, a typical requirement is computational efficiency. Kalman filter techniques are characterized by low computational cost and, being based on a solid statistical model, by a good robustness level. Their use for background updating was firstly proposed in [2]. Some modifications have been recently presented in order to better manage slow illumination variations [3]. This method can adapt to gradual changes but fails in cases of sudden change. In this paper we propose a different use of the Kalman filter paradigm and present a background updating algorithm able to deal with both gradual and sharp global illumination changes. Moreover, we address two problems that can complicate the functioning of predictive filters: the estimation of the measurement noise variance and the management of saturated pixels. Section 2 describes the use of the Kalman filter schema for background updating, Section 3 motivates and illustrates our method. Experimental results are described in Section 4 and Section 5 concludes the paper.

2 Kalman Filter for Background Updating

The Kalman filter [1,5] is an optimal estimator of the state of processes which satisfies: (a) they can be modeled by a linear system, (b) the measurement and the process noise are *white*, and have zero mean gaussian distributions. Under these conditions, knowing the input (external controls u_t) and the output (measurements z_t) of the system, the Kalman filter provides an optimal estimate of the state of the process (x_t), by minimizing the variance of the estimation error and constraining the average of the estimated outputs and the average of the measures to be the same. It is characterized by two main equations: the state equation (1) and the measurement equation (2):

$$x_t = Ax_{t-1} + Bu_{t-1} + w_{t-1} \tag{1}$$

$$z_t = Cx_t + v_t \tag{2}$$

A is the state transition matrix, B is the external control transition matrix, w represents the process noise, C is the transition matrix that maps the process state to the measurement, and v represents the measurement noise. The Kalman filter works in two steps: prediction and correction steps. The former uses the state of the system and the external control at time $t-1$ to predict the current state ($\hat{x}_t^-$), the latter uses the current measure z_t to correct the state estimation ($\hat{x}_t$). The working schema of the Kalman filter is illustrated in Figure 1. The factor K_t, the *gain of the filter*, is chosen in order to minimize the variance of the estimate error (P_t). The difference between the measure and the state predicted value ($z_t - C\hat{x}_t^-$) is called *innovation*.

The application of Kalman theory to the background updating task, e.g. [2] [4], typically considers the temporal distribution of the intensity levels of each pixel p and models it with a Kalman filter: the state of the system is the background value of pixel $x_t(p)$, and the measurement at time t is the value $I_t(p)$ of the pixel in the current image. The system input term (u_t) is set to zero, and the

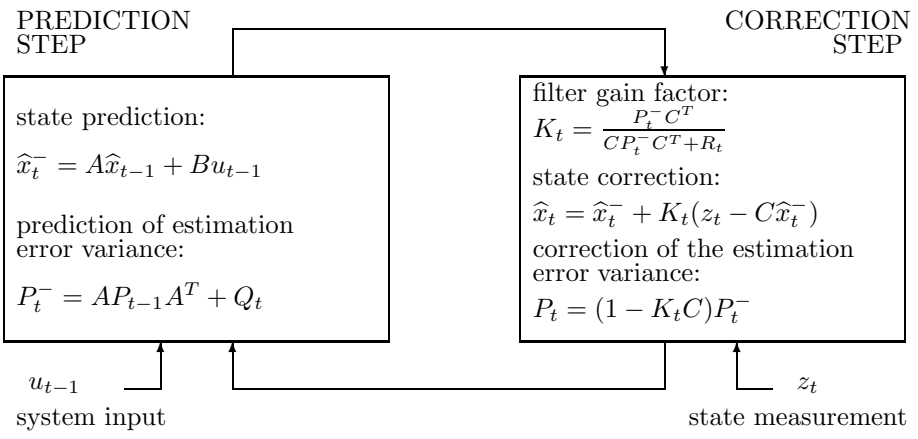


Fig. 1. The general schema of a Kalman filter. R_t and Q_t represent the variances of the gaussian random variables describing, respectively, the measurment noise and the process noise: $v_t = N(0, R_t)$ and $w_t = N(0, Q_t)$.

temporal distributions of the background intensities are considered constant. All unexpected changes are described by the random noise w_t, which by hypothesis is a zero mean gaussian variable. In order to prevent foreground pixels modifying the background image, a different gain factor is introduced if the innovation overcomes a given threshold. In this approach, gradual illumination changes can be captured only by the random noise term (w_t) which has to vary in time according to them. In this way the zero mean hypothesis could be, in principle, violated. Moreover, sudden illumination changes cause intensity variations that are considered as foreground pixels and cannot be correctly managed. In [3] a new term is introduced to model the intensity variations caused by gradual illumination changes. The random noise term is split into a slow varying component (β_t) and a zero mean gaussian component (μ_t). In this way the state of the system consists of two components: the background value (x_t) and its changing rate (β_t) whose role is to model gradual illumination variations. In order to deal with foreground objects the authors introduce a statistical test which takes into consideration the innovation value. If the innovation is below a given threshold they say that the measure accords with the prediction and the filter works as usual (the filter is open), otherwise a different updating criterion is adopted (the filter is closed). The threshold, computed on a statistical base, is related to the sum of the estimate error variance (P) and the measurement noise variance (R).

3 The Proposed Background Updating Module

We argue that it is incorrect to model all variations in the background intensities as noise added to the process. Variations caused by global illumination changes are external events that are logically different from variations due to the presence

of foreground objects. We develop a module that measures such global changes, and use this information as an external input to the system. The module exploits the fact that global illumination changes affect in a consistent way all the pixels in the image while variations due to foreground objects involve only a subset of them. In the subsequent paragraphs, the three main contributions of our paper are detailed, followed by the final schema of the proposed Kalman filter.

Estimation of global illumination changes. Following the light reflection model introduced by [6], the intensity at a given point on an object is the product of the illumination and a shading coefficient which is a characteristic of the object. A global illumination variation causes a modification of the pixel values that is approximately proportional to the preceding ones. Although this is an oversimplification, the introduced errors are generally acceptable and they can be modeled inside the Kalman filtering framework. To detect global illumination changes we analyze the distribution of the ratios $I_t(p)/x_{t-1}(p)$, for each pixel p. If the illumination is stable the distribution has a main peak around the value 1 and only foreground pixels contribute to the distribution tails. When a change takes place the main peak shifts towards values greater than 1 if the illumination increases, lower than 1 otherwise. Under the hypothesis that the majority of pixels belong to the background, the median of the distribution of ratios provides a global estimation of the changing factor. We verify experimentally that a single factor is not adequate for all tbackground pixels, but slight local variations permit the system to accommodate for the non uniform reflection characteristics of different image locations. These local variations with respect to the global changing factor have to be small enough to prevent foreground pixels to be detected as background. This is controlled by a threshold ($K_{thr} = 1.02$ in the experiments). The estimation procedure is reported in Algorithm 1.

Estimation of the measurement noise variance. The parameter R plays a crucial role in the functioning of the filter. Generally, it is assumed independent of time (t) and pixel position (p) and is estimated by analysing a portion of the input sequence. After running the background updating module on several sequences, acquired by devices (camera/lenses) with different characteristics and encoded in different ways, we noted a non homogeneous behaviour in correspondence of dark and bright pixels. We verified that one cause was the inaccurate estimation of the measurement error variance. To reduce this effect, we estimate the parameter R considering the intensity level ℓ of the pixel p. The estimation procedure is reported in Algorithm 2 ($N = 400$ and $L = 256$ the experiments).

Management of saturated pixels. Saturated pixels have the maximum intensity level L. Their actual intensity is unknown as all intensities above the maximum are mapped to L. Let us consider a background image where some pixels have the value L and after an increase of illumination a number of new pixels assume the value L. When the illumination comes back to the previous level, all the pixels in the image reduce their intensity levels by approximatively the same factor. As a result all of the saturated pixels receive incorrectly the

Algorithm 1 Estimate of the global illumination change

divide image I_t into a rectangular grid r_{ij} $(i = 1 \dots N, j = 1 \dots M)$
for each rectangle r_{ij} **do**
  $D_{ij} \leftarrow$ distribution of ratios $I_t(p)/x_{t-1}(p)$ for all pixel $p \in r_{ij}$ such that:
  $(x_{t-1}(p) \neq 0) \wedge (x_{t-1}(p) < L) \wedge (I(p) < L)$ {deal with saturated pixel}
  $k_t(ij) \leftarrow$ median of D_{ij}
end for
$K \leftarrow$ median of $k_t(ij)$ over all rectangles r_{ij}
for each rectangle r_{ij} **do**
  $k_t(ij) \leftarrow$ *unknown* if $(k_t(ij)/K > K_{thr}) \vee (k_t(ij)/K < 1/K_{thr})$
end for
for each rectangle r_{ij} such that $k_t(ij) =$ *unknown* **do**
  $k_t(ij) \leftarrow$ average of K and $k_t(uv)$, with r_{uv} adjacent to r_{ij} and $k_t(uv) \neq$ *unknown*
end for
for all pixels p **do**
  let r_1, r_2, r_3, r_4 be the 4 rectangles closest to p
  let w_1, w_2, w_3, w_4 be the distance from p to the rectangles centers
  $k_t(p) \leftarrow (\sum_{h=1}^{4} k_t(h) w_h) / \sum_{h=1}^{4} w_h$
end for
return the map $k_t(p)$

Algorithm 2 Estimate of the measurement noise variance

consider the image sequence $I_1(p) \dots I_N(p)$
let L be the maximum quantization level of the illumination intensity
for each level $\ell \in [0, L]$ **do**
  build an histogram H_ℓ and initialize it to 0
end for
for all image pair (I_k, I_{k+1}) with $k = 1 \dots N - 1$ **do**
  for all pixels p **do**
    $\ell \leftarrow I_k(p)$
    increment $H_\ell[I_{k+1}]$
  end for
end for
for each level $\ell \in [0, L]$ **do**
  $R_\ell \leftarrow$ median of the H_ℓ portion between its 10^{th} and 90^{th} percentile
  $R_\ell \leftarrow R_\ell / \sqrt{2}$
end for
smooth R_ℓ with a mean filter of length $w \simeq 0.1 * L$
$m \leftarrow \arg\max(R_\ell)$
for each level from $m + 1$ to L **do**
  $R_\ell \leftarrow R_m$
end for
return the vector R

Table 1. $\widehat{x}_t$ estimate of the background value; $\widehat{x}_t^-$ *a priori* estimate of the background value; $P_{g,t}$ estimation error variance; $P_{g,t}^-$ *a priori* estimation error variance; $K_{g,t}$ gain factor of the filter when the measure is compatible with the prediction (the filter is open); $K'_{g,t}$ gain of the filter when there is incompatibility between prediction and measure (the filter is closed); z_t is the current measure; α controls the noise of the k_t estimation process; γ controls the test for establishing if the prediction and the measure are compatible (if to open or to close the filter); ϱ controls the gain of the closed filter.

Initialization
Estimate initial background image $x_0(p)$
Estimate the measurement noise variance R_ℓ (Algorithm 2)
Initialize a map of labels: $L(p) \leftarrow \begin{cases} unknown & \text{if } x_0(p) = L \\ known & \text{otherwise} \end{cases}$
Initialize a map of counters: $C(p) \leftarrow 0$

For each frame t
Estimate the illumination change factors map $k_t(p)$ (Algorithm 1)

For each pixel p		
$\hat{x}_t^- \leftarrow k_t x_{t-1}$ $\ell \leftarrow Range_0^L(\hat{x}_t^-)$ $P_{\ell,t}^- \leftarrow (\alpha\|1-k_t\|+1)P_{\ell,t-1}^-$	$\Delta \leftarrow z_t - \hat{x}_t^-$ $\eta_{\ell,t} \leftarrow \gamma\sqrt{P_{\ell,t}^- + R_\ell}$	Prediction step

unknown	*true*	*true*	$\hat{x}_t = z_t$ increment $C(p)$	Correction step
		$z_t < L - \eta_{\ell,t}$		
		false		
	$C(p) \le 0$	*true*	increment $C(p)$ if $(C(p) > C_{thr})$ $L(p) = known$ $K_{\ell,t} = \frac{P_{\ell,t}^-}{P_{\ell,t}^- + R_\ell}$ $\hat{x}_t = \hat{x}_t^- + K_{\ell,t}\Delta$ $P_{\ell,t} = (1 - K_{\ell,t})P_{\ell,t}^-$	
	false	$\Delta^2 < \eta_{\ell,t}$		
		false	decrement $C(p)$ if $(C(p) = 0)$ $\hat{x}_t = L$	
$L(p)$		*true*	$K_{\ell,t} = \frac{P_{\ell,t}^-}{P_{\ell,t}^- + R_\ell}$ $\hat{x}_t = \hat{x}_t^- + K_{\ell,t}\Delta$ $P_{\ell,t} = (1 - K_{\ell,t})P_{\ell,t}^-$	
	true	$\Delta^2 < \eta_{\ell,t}$		
known		*false*	$K'_{\ell,t} = \varrho\frac{P_{\ell,t}}{P_{\ell,t}+\Delta^2}$ $\hat{x}_t = \hat{x}_t^- + K'_{\ell,t}\Delta$	
	$\hat{x}_t^- < L$	*true*	$K'_{\ell,t} = \varrho\frac{P_{\ell,t}}{P_{\ell,t}+\Delta^2}$ $\hat{x}_t = \hat{x}_t^- + K'_{\ell,t}\Delta$	
	false	$z_t < L - \eta_{\ell,t}$		
		false	$\hat{x}_t = \hat{x}_t^-$	

same value, and the information about the different values before the illumination variations are lost. We introduce a modification to the background updating schema in order to distinguish between pixels with a *known* or *unknown* intensity level. Initially, all of the unsaturated pixels are labelled *known* while the saturated ones are labelled *unknown*. They can move to the *known* state if their background value, after an illumination reduction, remains unsaturated for a sufficiently long period (controlled by a threshold C_{thr}). The resulting background updating schema is reported in Table 1.

4 Experimental Results

The proposed algorithm has been tested within a vison-based traffic control system named SCOCA [7]. Its purpose is the counting and classification of vehicles crossing a road intersection observed by a camera. The detection of vehicles relies on a background differencing technique. Hence the performance of our algorithm compared to that of [3] was measured by observing the output of SCOCA. We used three image sequences coming from different intersections. Sequences $S1$ and $S2$ are composed of frames taken from a standard color surveillance camera, endowed with autoiris lens, and compressed into *mpeg* files. $S3$ is composed of gray scale frames taken from a progressive scan camera with electronic shutter, each one compressed in a *jpg* file. $S1$ and $S2$ contain gradual illumination changes both due to real lighting variations and the activation of the autoiris lenses. $S3$ contains some sharp illumination changes due to the electronic variation of the shutter, and a significant presence of saturated pixels. The table in Figure 2 reports the performance of SCOCA in terms of correct detection rate and correct classification rate. For the sequences $S1$ and $S2$, the use of the proposed algorithm provides better performances showing its superiority in detecting foreground objects and in localizing their boundaries, as suggested by the improvements in the classification scores. Considering the third sequence a

	Detection rate		Classification rate	
	alg. [3]	our	alg. [3]	our
S1	90.7%	91.4%	85.4%	87.6%
S2	75.5%	77.1%	70.6%	74.8%
S3	–	78.5%	–	73.8%

Fig. 2. Comparison of SCOCA performances using the two algorithms

Fig. 3. From left to right: a frame in the middle of sequence S3; the corresponding backgrounds computed by alg. [3] and ours; the foreground pixels

direct comparison is not possible because only the proposed algorithm provides a reasonable result, the other one went into an inconsistent state after the first sharp illumination change (see Figure 3).

5 Conclusions

We have presented a new algorithm for background updating based on Kalman filtering technique, which is robust to gradual and sharp illumination changes. The most significant novelty we have introduced, which make the algorithm robust to gradual and sharp illumination changes, is the estimation of the global illumination variations and its use as an external control of the Kalman filter. The experiments have shown that the algorithm extends the range of sequences where it can work successfully. Its major limitation, i.e. inability to manage dynamic background pixels, is the subject of our current research.

References

1. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME Journal of Basic Engineering, 82 (Series D):35–45, 1960.
2. Karman, K., Brandt, A., Gerl, R.: Moving object segmentation based on adaptive reference images. In Signal Processing V: Theories and Applications, pp 951–954, Barcelona, Spain, September 1990.
3. Boninsegna, M., Bozzoli, A.: A Tunable Algorithm to Update a Reference Image. Signal Processing: Image Communication, 16(4) pp 353–365, November 2000.
4. Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., Russell, S.: Toward robust automatic traffic scene analyis in real-time. Proc. Int. Conf. Pattern Recognition, pp 126–131, 1994
5. Catlin. D.E.: Estimation, control, and the discrete Kalman filter. Springer-Verlag, 1989
6. Oppenheim, A.V., Schafer, R.W., Stockham Jr, T.G.: Nonlinear filtering of multiplied and convolved signals. Proc. IEEE, vol. 56, pp 1264–1291, August 1968
7. Messelodi, S., Modena, C.M., Zanin, M.: A computer vision system for the detection and classification of vehicles at urban road intersections. ITC-irst Technical Report T04-02-07, February 2004
8. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, P.: Pfinder: Real-time tracking of human body. IEEE Transaction PAMI, vol.19, pp 780–785, July 1997
9. Grimson, W.E.L., Stauffer, C.: Adaptive background mixture models for real-time tracking. Proc. IEEE Computer Vision Pattern Recognition, vol.1, pp 22–29, 1999
10. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric kernel density estimation for video surveillance. Proc. of the IEEE, vol.90, pp 1151–1163, N.7, July 2002
11. Li, L., Huang. W., Gu, I., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. IEEE Trans. Image Processing, vol.13, Nov. 2004
12. Li, L., Leung, M.: Integrating intensity and texture differences for robust change detection. IEEE Trans. Image Processing, vol.11, pp 105–112, February 2002
13. Wixson, L.: Detecting salient motion by accumulating directionary-consistent flow. IEEE Trans. PAMI, vol.22, pp 269–274, August 2000

Greedy Algorithm for Local Contrast Enhancement of Images

Kartic Subr, Aditi Majumder, and Sandy Irani

School of Information and Computer Science, University of California, Irvine

Abstract. We present a technique that achieves local contrast enhancement by representing it as an optimization problem. For this, we first introduce a scalar objective function that estimates the *average local contrast* of the image; to achieve the contrast enhancement, we seek to maximize this objective function subject to strict constraints on the local gradients and the color range of the image. The former constraint controls the amount of contrast enhancement achieved while the latter prevents over or under saturation of the colors as a result of the enhancement. We propose a greedy iterative algorithm, controlled by a single parameter, to solve this optimization problem. Thus, our contrast enhancement is achieved without explicitly segmenting the image either in the spatial (multi-scale) or frequency (multi-resolution) domain. We demonstrate our method on both gray and color images and compare it with other existing global and local contrast enhancement techniques.

1 Introduction

The problem of contrast enhancement of images enjoys much attention; its applications span a wide gamut, ranging from improving visual quality of photographs acquired with poor illumination [1] to medical imaging [2].

Common techniques for global contrast enhancements like global stretching and histogram equalization do not always produce good results, especially for images with large spatial variation in contrast. A number of local contrast enhancement methods have been proposed to address exactly this issue. Most of them explicitly perform image segmentation either in the spatial(multi-scale) or frequency(multi-resolution) domain followed by a contrast enhancement operation on each segment. The approaches differ mainly in the way they choose to generate the multi-scale or multi-resolution image representation (anisotropic diffusion [2], non-linear pyramidal techniques[3], multi-scale morphological techniques [4,5], multi-resolution splines [6], mountain clustering [7], retinex theory [8,9]) or in the way they enhance contrast after segmentation (morphological operators [5], wavelet transformations [10], curvelet transformations [11], k-sigma clipping [8,9], fuzzy logic [12,7], genetic algorithms [13]).

In this paper we present a local contrast enhancement method driven by a scalar objective function that estimates the local average contrast of an image. Our goal is to enhance the local gradients, which are directly related to the local contrast of an image. Methods that manipulate the local gradients[14,15] need

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 171–179, 2005.

to integrate the manipulated gradient field to construct the enhanced image. This is an approximately invertible problem that requires solving the poisson equation dealing with differential equations of potentially millions of variables.

Instead, we achieve gradient enhancement by trying to maximize a simple objective function. The main contributions of this paper are:

- a simple, scalar objective function to *estimate* and *evaluate* the average local contrast of an images,
- an efficient greedy algorithm to enhance contrast by seeking to maximize the above objective function.

We present our contrast enhancement algorithm for gray image in Section 2. In Section 3 the extension of this method to color images is described, followed by the results in Section 4. Finally, we conclude with future work in Section 5.

2 Contrast Enhancement of Gray Images

2.1 The Optimization Problem

The perception of contrast is directly related to the local gradient of an image [16]. Our objective is to enhance the local gradients of an image subject to strict constraints that prevent both over/under-saturation and unbounded enhancement of the gray values. Thus, we propose to maximize the objective function

$$f(\Omega) = \frac{1}{4|\Omega|} \sum_{p \in \Omega} \sum_{q \in N_4(p)} \frac{I'(p) - I'(q)}{I(p) - I(q)} \tag{1}$$

subject to the constraints,

$$1 \leq \frac{I'(p) - I'(q)}{I(p) - I(q)} \leq (1 + \delta) \tag{2}$$

$$L \leq I'(p) \leq U \tag{3}$$

where scalar functions $I(p)$ and $I'(p)$ represent the gray values at pixel p of the input and output images respectively, Ω denotes set of pixels that makes up the image, $|\Omega|$ denotes the cardinality of Ω, $N_4(p)$ denotes the set of four neighbors of p, L and U are the lower and upper bounds on the gray values (eg. $L = 0$ and $U = 255$ for 8 bit gray values between 0 and 255), and $\delta > 0$ is the single parameter that controls the amount of enhancement achieved. We seek to maximize our objective function by pronouncing the local gradient around a pixel in the input image to the maximum possible degree. The constraint defined by Equation 2 assures a bounded enhancement of gradients. The lower bound ensures that the signs of the gradients are preserved and that the gradients are never shrunk. The upper bound ensures a bounded enhancement of contrast controlled by the parameter δ. The constraint defined by Equation 3 ensures that the output image does not have saturated intensity values.

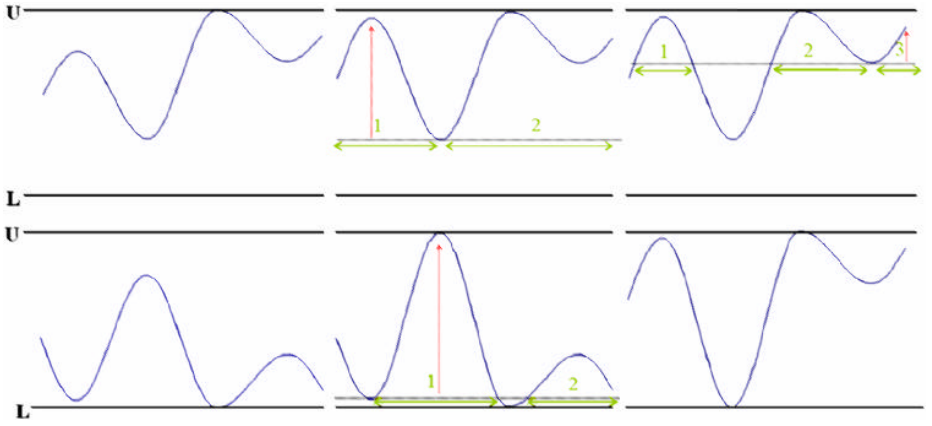


Fig. 1. Graphs showing key steps in our algorithm when applied to a 1D signal. Hillocks formed at each stage are numbered. Their extent is shown with green arrows and enhancement with red arrows. *Top Row*(left to right): Input signal; sweep plane through first minima. Note that hillock 1 is pushed up as much as possible so that the δ constraint is not violated while hillock 2 cannot be enhanced at all since it will violate the saturation constraint; sweep plane through second minima. Note that hillocks 1 and 2 cannot be enhanced. However hillock 2 from the previous step has split into 2 and 3 of which the latter can be enhanced. *Bottom Row*(left to right): Invert Signal from previous step; sweep plane through first minima in the inverted signal; output signal obtained by re-inverting.

2.2 The Algorithm

We design a greedy algorithm to solve the optimization problem in 2.1. Our algorithm is based on the fundamental observation that given two neighboring pixels with gray values r and s, $s \neq r$, scaling them both by a factor of $(1+\delta)$ results in r' and s' such that

$$\frac{r'-s'}{r-s} = (1+\delta) \tag{4}$$

Thus if we simply scale the values $I(p), \forall p \in \Omega$, by a factor of $(1+\delta)$, we obtain the maximum possible value for $f(\Omega)$. However, this could cause violation of Equation 3 at p, leading to saturation of intensity at that point. To avoid this, we adopt an iterative strategy, employing a greedy approach at each iteration.

Let us visualize the image I as a height-field sampled at the grid points of a $m \times n$ uniform grid. This set of samples represents Ω for a $m \times n$ rectangular image. Thus, the height at every pixel $p \in \Omega$, $I(p)$, is within L and U. A one dimensional example of the algorithm is shown in Figure 1.

For each iteration, we consider b, $L \leq b \leq U$. Next, we generate an $m \times n$ matrix R by marking the regions of I which are above the plane b as

$$R(i,j) = \begin{cases} 1 \text{ if } I(i,j) > b \\ 0 \text{ if } I(i,j) \leq b \end{cases} \tag{5}$$

(a) (b) (c)

Fig. 2. The original gray image (a), enhanced gray image using δ of 0.3 (b) and 2 (c). Note that parts of the image that have achieved saturation for $\delta = 0.3$ do not undergo anymore enhancement or show any saturation artifact for higher delta of 2. Yet, note further enhancement of areas like the steel band on the oxygen cylinder, the driver's thigh and the pipes in the background. The same method applied to the red, green and blue channel of a color image - the original image (d), enhanced image using δ of 0.4 (e) and 2 (f) - note the differences in the yellow regions of the white flower and the venation on the leaves are further enhanced.

In the example in Figure 1 these are the points above the green line. Next, we find the non-zero connected components in R, and label them uniquely. Let us call each such component, h_i^b, a *hillock*; i denotes the component number and b denotes the thresholding value used to define the hillocks. In Figure 1, the hillocks are shown numbered. Next, these hillocks are pushed up such that no pixel belonging to the hillock has the gradient around it enhanced by a factor more than $(1 + \delta)$ and no pixel is pushed up beyond U.

Our method involves iteratively sweeping threshold planes from L through U and greedily scaling the hillocks respecting the constraints at each sweep. Note that as we sweep successive planes, a hillock h_i^b can split into h_j^{b+1} and h_k^{b+1} or remain unchanged. But, two hillocks h_i^b and h_j^b can never merge to form h_k^{b+1}. This results from the fact that our threshold value increases from one sweep to the next and hence the pixels examined in an iteration are the subset of the pixels examined in the previous iterations. This observation helps us to perform two important optimizations. First, we obtain the new hillocks by only searching amongst hillocks from the immediately preceding sweep. Second, we store information about how much a hillock has been scaled as one floating point value per hillock. By ensuring that this value never exceeds $(1 + \delta)$ we avoid explicitly checking for gradient constraints at each pixel in each iteration. We omit these optimizations in the pseudo-code shown in Figure 3 for simplicity.

For low values of b, enhancement achieved on hillocks might not be $(1 + \delta)$ because of the increased chances of a peak close to U for large sized hillocks. As b increases, the large connected components are divided so that the smaller hillocks can now be enhanced more than before(see top row of Fig 1). This constitutes the upward sweep in our algorithm which enhances only the local hillocks of I and the image thus generated is denoted by I_1.

Algorithm *Enhance*(δ, I, L, U)
Input: Control parameter δ
Input Image I
Lower and upper bounds L and U
Output: Enhanced Image I'
Begin
Find $P = \{b|b = I(p)\}, \forall p$ at minimas/saddle points
Sort P into list P'
$I' \leftarrow I$
$I' \leftarrow$ *SweepAndPush*(I', P', δ)
$I' \leftarrow U - I'$
Find $Q = \{b|b = I'(p)\}, \forall p$ at minimas/saddle points
Sort Q into List Q'
$I' \leftarrow$ *SweepAndPush*(I', Q', δ)
$I' \leftarrow U - I'$
Return I'
End

Algorithm *SweepAndPush*(I, S, δ)
Input: Input image I
List of gray values S
Control parameter δ
Output: Output image I'
Begin
$I' = I$
for each $s \in S$
obtain boolean matrix $B \ni B_{ij} = 1$ iff $I_{ij} \geq s$
Identify set of hillocks H in B
for each $H_i \in H$
find $p_{max} \ni I(p_{max}) \geq I(p) \ \forall \ p, p_{max} \in H_i$
$\delta_{max} = min(\delta, (U - s)/(I(p_{max}) - s) - 1.0))$
for each $p \in H_i$
$\delta_{apply} = \delta_{max}$
lookup $\delta_h(p)$, the net scaling factor in the history for p
if $(\delta_{apply} + \delta_h > \delta)$
$\delta_{apply} = \delta - \delta_h$
$I'(p) = (1 + \delta_{apply}) * (I(p) - s) + s$
update $\delta_h = \delta_h + \delta_{apply}$
End

Fig. 3. The Algorithm

Further enhancement can be achieved by enhancing the local *valleys* also. So, the second stage of the our method applies the same technique to the complement of $I_1 = U - I_1(p)$ to generate the output I_2. I_2 is complemented again to generate the enhanced output image $I' = U - I_2(p)$(see bottom row of Fig 1).

We perform $U - L$ sweeps to generate each of I_1 and I_2. In each sweep we identify connected components in a $m \times n$ matrix. Thus, the time-complexity of our algorithm is theoretically $O((U - L)(mn + log(U - L)))$. The logarithmic term arises from the need to sort lists P and Q (see Fig. 3) and is typically dominated by the mn term. However, we observe that hillocks split at local points of minima or saddle points. So, we sweep planes only at specific values of b where some points in the height field attain a local minima or saddle point. This helps us to achieve an improved time complexity of $O(s(mn + log(s)))$ where s is the number of planes swept (number of local maximas, local minimas and saddle points in the input image). These optimizations are incorporated in the pseudocode of the algorithm in the Figure 3.

3 Extension to Color Images

One obvious way to extend the algorithm presented in Section 2.2 to color images is to apply the method independently to three different color channels. However, doing this does not assure hue preservation and results in hue shift, especially with higher values of δ, as illustrated in Figure 4. To overcome this problem we apply our method to the luminance component of the image only. We first linearly transform the RGB values to CIE XYZ space [17] to obtain the luminance (Y) and the chromaticity coordinate ($x = \frac{X}{X+Y+Z}$ and $y = \frac{Y}{X+Y+Z}$), and then apply our method only to Y keeping x and y constant, and finally convert the image back to the RGB space. However, in this case, the constraint in Equation 3 needs to be modified so that the resulting enhanced color lies within the

(a) (b) (c)

Fig. 4. Results of applying the algorithm from Section 2.2 on the red, green and blue channels separately. (a) Original image (b) Channels enhanced with a δ of 0.4 (c) Channels enhanced with a δ of 2. Note the artifacts on the petals of the yellow flower and on the leaf venation.

color gamut of the display device. Here we describe the formulation for color images.

The primaries of the display device are defined by three vectors in the XYZ color space $\overrightarrow{R} = (X_R, Y_R, Z_R)$, $\overrightarrow{G} = (X_G, Y_G, Z_G)$ and $\overrightarrow{B} = (X_B, Y_B, Z_B)$. The transformation from RGB to XYZ space is defined by a 3×3 matrix whose rows correspond to $\overrightarrow{R}$, $\overrightarrow{G}$ and $\overrightarrow{B}$. Any color in the XYZ space that can be expressed as a convex combination of $\overrightarrow{R}$, $\overrightarrow{G}$ and $\overrightarrow{B}$ is within the color gamut of the display device. Note that scaling a vector in the XYZ spaces changes its luminance only, keeping the chrominance unchanged. This is achieved by scaling Y while keeping (x, y) of a color constant. In addition, to satisfy the saturation constraint, we assure that the enhanced color lies within the parallelopiped defined by the convex combination of $\overrightarrow{R}$, $\overrightarrow{G}$ and $\overrightarrow{B}$.

Thus, the color at pixel p, given by $C(p) = (X, Y, Z)$ is to be enhanced to $C'(p) = (X', Y', Z')$ by enhancing Y to Y' such that the objective function

(a) (b) (c)

Fig. 5. (a) Original image, (b) algorithm from Sec. 2.2 applied with $\delta = 2$ on the red, green and blue channels separately. Note the significant hue shift towards purple in the stairs, arch and wall and towards green on the ledge above the stairs, (c) Using the method described in Sec.3 with $\delta = 2$ (separating the image into luminance and chrominance and applying the method to the former). Note that hue is preserved.

$$f(\Omega) = \frac{1}{4|\Omega|} \sum_{p \in \Omega} \sum_{q \in N_4(p)} \frac{Y'(p) - Y'(q)}{Y(p) - Y(q)} \tag{6}$$

is maximized subject to a perceptual constraint

$$1 \leq \frac{Y'(p) - Y'(q)}{Y(p) - Y(q)} \leq (1 + \delta) \tag{7}$$

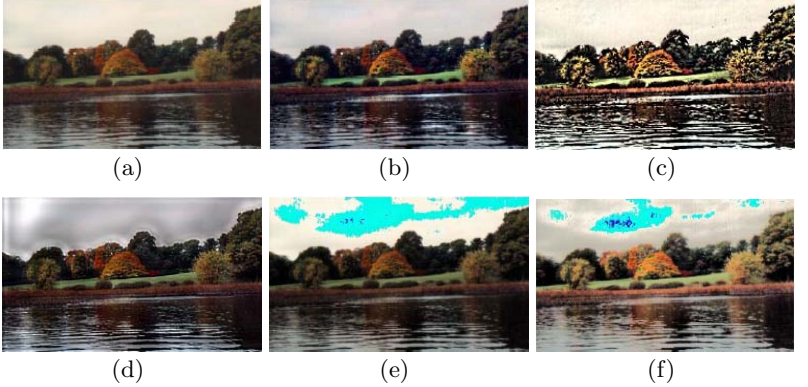
and a saturation constraint

$$(X', Y', Z') = c_R \overrightarrow{R} + c_G \overrightarrow{G} + c_B \overrightarrow{B}, \qquad 0.0 \leq c_R, c_G, c_B \leq 1.0 \tag{8}$$

Thus by changing just the saturation constraint, we achieve contrast enhancement of color images without saturation artifacts (Figure 5).

(a) (b) (c) (d)

Fig. 6. This figure compares our method with existing methods. (a) The original image, (b) our greedy-based method($\delta = 2$), (c) curvelet transformation [11], (d) method based on multi-scale retinex theory [9]. Note that (c) leads to a noisy image while (d) changes the hue of the image significantly



(a) (b) (c)

(d) (e) (f)

Fig. 7. (a) The original image, (b) our method, (c) multi-scale morphology method [5] - note the saturation artifacts that gives the image an unrealistic look, (d) Toet's method of multi-scale non-linear pyramid recombination [3] - note the halo artifacts at regions of large change in gradients, (e) global contrast stretching, (f) global histogram equalization - both (e) and (f) suffer from saturation artifacts and color blotches.

4 Results

We show the effect of our algorithm on gray images, with varying values of the input parameter δ (Figure 2). We also show the difference between directly applying the algorithm to the three color channels (Figure 4) and our adaptation in Section 3 to ensure color gamut constraints (Figure 5). Figures 6 and 7 compare our method with other local and global contrast enhancement methods.

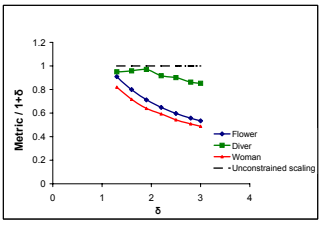


Fig. 8. Plot of $\frac{\alpha}{1+\delta}$ vs. $1+\delta$. The former is the average local contrast achieved while the latter is the average local contrast that would be achieved by scaling the image by $1+\delta$ without respecting constraints

Note that the objective function defined in Equation 1 or Equation 6 can also be used as an estimate of the average local contrast of an image, and hence, to evaluate the degree of enhancement achieved. According to the function, the maximum average local contrast that can be achieved *without any constraints* is given by $1+\delta$. However, imposing the constraints leads to a more practical average contrast value $\alpha < (1+\delta)$; as δ increases, the gap between $1+\delta$ and α widens since more pixels reach saturation and thus cannot achieve values close to $1+\delta$. Figure 8 plots the ratio $\frac{\alpha}{1+\delta}$ with $1+\delta$.

5 Conclusion and Future Work

We design a scalar objective function to describe the average local contrast of an image that has the potential to be used for estimating and evaluating the contrast of the image. We formulate the contrast enhancement problem as an optimization problem that tries to maximize the average local contrast of the image in a controlled fashion without saturation of colors. We present an efficient greedy algorithm controlled by a single input parameter δ to solve this optimization. Currently we are exploring adaptation of the same algorithm for tone mapping of high dynamic range images by changing δ spatially. Using the same concept, we can achieve seamless contrast enhancement of a selective region of interest in an image by varying the parameter δ smoothly over that region. Since our method works without explicitly segmenting the image either in the spatial or the frequency domain, it is very efficient. We are trying to exploit this efficiency to extend this method to video sequences, for which maintaining temporal continuity is of great importance.

References

1. Oakley, J.P., Satherley, B.L.: Improving image quality in poor visibility conditions using a physical model for contrast degradation. IEEE Transactions on Image Processing **7** (1998) 167–179
2. Boccignone, G., Picariello, A.: Multiscale contrast enhancement of medical images. Proceedings of ICASSP (1997)
3. Toet, A.: Multi-scale color image enhancement. Pattern Recognition Letters **13** (1992) 167–174
4. Toet, A.: A hierarchical morphological image decomposition. Pattern Recognition Letters **11** (1990) 267–274
5. Mukhopadhyay, S., Chanda, B.: Hue preserving color image enhancement using multi-scale morphology. Indian Conference on Computer Vision, Graphics and Image Processing (2002)
6. Burt, P.J., Adelson, E.H.: A multiresolution spline with application to image mosaics. ACM Transactions on Graphics **2** (1983) 217–236
7. Hanmandlu, M., Jha, D., Sharma, R.: Localized contrast enhancement of color images using clustering. Proceedings of IEEE International Conference on Information Technology: Coding and Computing (ITCC) (2001)
8. Munteanu, C., Rosa, A.: Color image enhancement using evolutionary principles and the retinex theory of color constancy. Proceedings 2001 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing XI (2001) 393–402
9. Rahman, Z., Jobson, D.J., , Woodell, G.A.: Multi-scale retinex for color image enhancement. IEEE International Conference on Image Processing (1996)
10. Velde, K.V.: Multi-scale color image enhancement. Proceedings on International Conference on Image Processing **3** (1999) 584–587
11. Stark, J.L., Murtagh, F., Candes, E.J., Donoho, D.L.: Gray and color image contrast enhancement by curvelet transform. IEEE Transactions on Image Processing **12** (2003)
12. Hanmandlu, M., Jha, D., Sharma, R.: Color image enhancement by fuzzy intensification. Proceedings of International Conference on Pattern Recognition (2000)
13. Shyu, M., Leou, J.: A geneticle algorithm approach to color image enhancement. Pattern Recognition **31** (1998) 871–880
14. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. ACM Transactions on Graphics, Proceedings of ACM Siggraph **21** (2002) 249–256
15. Prez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Transactions on Graphics, Proceedings of ACM Siggraph **22** (2003) 313–318
16. Valois, R.L.D., Valois, K.K.D.: Spatial Vision. Oxford University Press (1990)
17. Giorgianni, E.J., Madden, T.E.: Digital Color Management : Encoding Solutions. Addison Wesley (1998)

Probabilistic Model-Based Background Subtraction

V. Krüger, J. Anderson, and T. Prehn

[1] Aalborg Media Lab, Aalborg University, Copenhagen
Lautrupvang 15, 2750 Ballerup
[2] Aalborg University Esbjerg, Niels Bohrs Vej 8, 6700 Esbjerg, Denmark

Abstract. In this paper we introduce a model-based background subtraction approach where first silhouettes, which model the correlations between neightboring pixels are being learned and where then Bayesian propagation over time is used to select the proper silhouette model and tracking parameters. Bayes propagation is attractive in our application as it allows to deal with uncertainties in the video data during tracking. We eploy a particle filter for density estimation. We have extensively tested our approach on suitable outdoor video data.

1 Introduction

Most vision systems work well in controlled environments, e.g., attempts to recognize humans by their face and gait has proven to be very successful in a lab environment. However, in uncontrolled environments, such as outdoor scenarios, the approaches disgracefully fail, e.g., the gait recognition drops from 99% to merely 30%. This is mainly due to low quality video data, the often small number of pixels on target and visual distractors such as shadows and strong illumination variations.

What is needed are special feature extraction techniques that are robust to outdoor distortions and that can cope with low-quality video data. One of the most common feature extraction techniques in surveillance applications is background subtraction (BGS) [5, 3, 8, 6]. BGS approaches assume a stable camera. They are able to learn a background as well as possible local image variations of it, thus generating a background model even of non-rigid background objects. During application the model is compared with novel video images and pixels are marked according to the belief that they are fitting the background model. It can be observed that BGS approaches are not able to distinguish between a foreground object and its shadow. And very often, the very same objects cause different outputs when the scenario changes: E.g. when a person walks on green grass or gray concret the outout can be severely different.

In this paper we present a Model-based Background Subtracting (MBGS) method that learns in addition to the background also the foreground model. A particle filter is used for finding and tracking the right silhouette.

In this work, without limit of generality, we consider only humans as objects and ignore objects that look different from humans. Also, we limit our discussion

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 180–187, 2005.

to silhouettes of humans as they deliver a fairly clothing-independent description of an individual.

The Model-based Background Subtraction System (MBGS System) consists of a learning part to learn possible foreground objects and a MBGS part, where the output of a classical BGS is verified using the previously trained foreground object knowledge.

To learn and represent the foreground knowledge (here silhouettes of humans) is non-trivial due to the absence of a suitable vector space. One possibility is to describe the data in a hierarchical manner, using a suitable metric and a suitable representation of dynamics between the silhouettes. Our approach for a hierarchical silhouette representation is inspired by [4, 13].

In the second part, we again consider the silhouettes as densities over spatial coordinates and use normalized correlation to compute the similarity between the silhouette density and the computed one in the input image. Tracking and silhouette selection is being done using Bayesian propagation over time. It can be applied directly since we are dealing with densities and it has the advantage that it considers the uncertainty in the estimation of the tracking parameters and the silhouette selection. The densities in the Bayesian propagation are approximated using an enhancement of the well-known Condensation method [7]. A similar enhancement of the Condensation method has been applied in video based face recognition [11].

The remainder of this paper is organized as follows: In Sec. 2 we introduce the learning approaches. The actual BGS method is discussed in Sec. 3. We conclude with experimental results in Sec. 4 and final remarks are in Sec. 5.

2 Learning and Representation of Foreground Objects

In order to make use of foreground model knowledge, our main idea is the following: Apply the classical BGS to a scenario that is controlled in a manner that facilitates the learning process. In our case, since we want to learn silhouettes of humans, that only humans are visible in the scene during training and that the background variations are kept as small as possible to minimize distortions. Then, we use this video data to learn the proper model knowledge.

After the application of a classical BGS, applying mean-shift tracking [1] allows to extract from the BGS output-data a sequence of small image patches containing, centered, the silhouette. This procedure is the same as the one used in [9], however, with the difference that here we do not threshold the BGS output but use probabilistic silhouettes (instead of binary ones as in [9]) which still contain for each silhouette pixel the belief of being a foreground pixel.

To organize this data we use, similar to [4], a combination of tree structuring and k-means clustering. We use a top down approach: The first level is the root of the hierarchy which contains all the exemplars. Then the second level is constructed by using a the k-means clustering to cluster the exemplars from the root. The third level is constructed by clustering each cluster from the second level, again, using k-means, see Fig. 1 for an example. The k-means clustering uses the Kullback-Leibler divergence measure which measures.

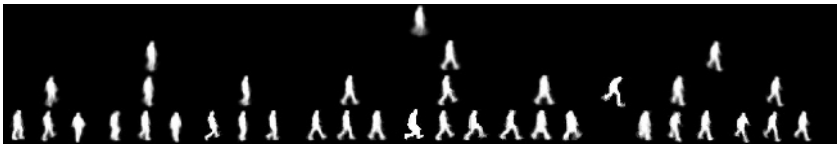


Fig. 1. An example of our clustering approach: 30 exemplars with K=3 and the algorithm stops after reaching 3 levels

Once the tree is constructed, we generate a Markov transition matrix: Assuming that the change over time from one silhouette to a next one can be understood as a first order Markov process, the Markov transition matrix M_{ij} describes the transition probability of silhouette s_j following after silhouette s_i at level l in the hierarchy. During MBGS application particle filtering [7, 10, 2] will be used to find the proper silhouette (see Sec. 3). The propagation of silhouettes over time is non-trivial, as silhouette do not form a vector space. However, what is sufficient, is a (not necessarily symmetric) metric space, i.e., given a silhouette s_i, all silhouettes are needed that are close according to a given metric. In the tree structure similar silhouettes are clustered which facilitates the propagation process. The Markov transition matrix M_{ij} on the other hand describes directly the transition likelihoods between clusters.

3 Applying Background Subtraction and Recognizing Foreground Objects

The MBGS system is built as an extension to a pixel based BGS approach. It uses foreground models to define likely correlations between neighbored pixels in the output $P(\mathbf{x})$ of the BGS application.

Each pixel in the image $P(\mathbf{x})$ contains a value in the range $[0, 1]$, where 1 indicates the highest probability of a pixel being a foreground pixel. A model in the hierarchy can be chosen and deformed according to a 4-D vector

$$\theta = [i, s, x, y], \tag{1}$$

where x and y denote the position of the silhouette in the image P, s its scale, and i is a natural number that refers to a silhouette in the hierarchy.

We use normalized correlation to compute the distance between a model silhouette, parameterized according to a deformation vector θ_t and the appropriate region of interest in the BGS image $P_t(\mathbf{x})$, appropriately normalized.

In order to find at each time-step t the most likely θ_t in the image $P_t(\mathbf{x})$, we use Bayesian propagation over time

$$\begin{aligned} & p(\theta_t|P_1, P_2, \ldots, P_t) \equiv p_t(\alpha_t, i_t) \\ & = \sum_{i_{t-1}} \int_{\alpha_{t-1}} p(P_t|\alpha_t, i_t) \\ & \qquad p(\alpha_t, i_t|\alpha_{t-1}, i_{t-1}) p_{t-1}(\alpha_{t-1}, i_{t-1}) \end{aligned} \tag{2}$$

with $\alpha_t = [s, x, y]_t$. Capital "P_t" denotes the probability images while little "p" denotes density functions. We approximate the posteriori density $p(\theta_t|P_1, P_2, \ldots, P_t)$ with a sequential Monte Carlo method [2, 7, 10, 12]. Using Bayesian propagation allows to take into account the uncertainty in the estimated parameter. Monte Carlo methods use random samples for the approximation of a density function. Our MBGS system uses separate sample sets for each object in the input image. A new sample set is constructed every time a new object in the video image matches sufficiently well.

As the diffusion density $p(\alpha_t, i_t|\alpha_{t-1}, i_{t-1})$ in Eq. 2 we use the Brownian motion model due to the absence of a better one. For the propagation of the position and scale parameters, x, y, and s, this is straight forward. The propagation of the silhouette is, however, not straight forward. Inspired by [11] we use the variable i to reference a silhouette in the silhouette database. By considering the joint distribution of the silhouette id with the tracking parameter we are able to view the tracking and the recognition as a single problem. By marginalizing over the geometric parameters $\alpha = (x\ \ y)$,

$$p(i_t|Z_1, \ldots, Z_t) = \int_{\alpha_t} p(\alpha_t, i_t|Z_1, \ldots, Z_t) \ . \tag{3}$$

we can estimate the likelihood of each silhouette at any time. In [11] where the parameter i is constant, it is sufficient for the recogntion to wait until all particles of the Monte Carlo Markov Chain have converged to the same identity. This is equivalent to minimizing the uncertainty, i.e., the entropy.

In this problem setup, however, the correct silhouette parameter i is not constant but changes as the person walks. Therefore, we have to consider the two following issues: (1) We have to find a likelihood measure $P(i_t|i_{t-1})$ for the propagation step: given a silhouette i_{t-1}, what are the likelihoods for the other silhouettes. (2) We have to define an approach that allowes to approximate the density p_t with an MCMC technique. This is complicated because one wants the particles to clearly converge to the correct silhouette at each time step and at the same time wants enough diffusion in the propagation step to allow silhouette changes.

Issue (1) was partially solved in the learning process (see Sec. 2) where silhouettes were clustered and the Markov transition matrix $M(i, j)$ was computed which represents the transition from silhouette i to silhouette j.Then,

- the likelihood for selecting a silhouette from a certain silhouette cluster in the hierarchy is computed from the Markov transition matrix M by marginalizing over the silhouettes in that particular cluster.
- Within a cluster, the new silhouette is then chosen randomly.

The reason for marginalizing over the clusters is because our training data is too little so that the Markov transition matrix M appeares to be specific to the training videos.

In order to approach issue (2), we diffuse only 50% of the particles at each time step with respect to the silhouette number. Our experiments have shown

that if the diffusion of the silhouette number was larger then there was often no clear convergence to one particluar silhouette. This, however, is needed to assure recognition of the correct silhouette at each time-step.

4 Experiments

In this section we present qualitative and quantitative results obtained from experiments with our MBGS implementation. The experiments clearly show the potentials of an effective MBGS approach. The purpose of the experiments was to verify that the drawbacks of the classical BGS approach, which were mentioned in section 1, can be remedied with MBGS. More specifically the MBGS system verifies the following: (1) Because shadows are not part of the model information provided, these will be classified as background by the implemented MBGS approach. In fact, most non-model object types will be classified as background, and therefore MBGS allows for effective object type filtering. (2) The output presented from the MBGS does not vary, even when the scenario changes significantly. If a model is presented as output, it is always presented intact. The object behind a silhouette is therefore always determinable.

Qualitative verification is done by comparing the AAU MBGS system with two previously developed pixel-based BGS approaches. One is the non-parametric approach developed at Maryland (UMD BGS) [3]. The other (AUE BGS), which utilizes alternative image noise filtering, has previously been developed at AUE.

Figure 2 shows a scenario, with a pedestrian walking behind trees, thereby at times being occluded. The output of two pixel-based approaches is shown in the lower part of the figure. Notice that the shadow cast by the pedestrian is classified as foreground by these pixel-based BGS approaches. Since the MBGS system operates by inserting a model as foreground, this problem is effectively resolved. Figure 3 shows the same scenario, in a frame where the pedestrian is heavily occluded. The occlusion causes the pedestrian to more or less disappear with pixel based approaches. This happens because the occlusion divides the pedestrian silhouette into separate smaller parts, which are then removed by the applied image filters. The scenario presented in figure 4, shows two pedestrians walking towards each other, thereby crossing behind lamp posts and a statue. When processing this scenario, a combination of image filtering and the background variation, renders the silhouettes of the pixel-based approaches unidentifiable. Also both pixel-based approaches severely distorts the silhouettes of the pedestrians. By only inspecting the pixel-based results, it is hard to tell that the foreground objects are actually pedestrians.

In a quantitative evaluation we have investigated the correctness of the particle method in matching the correct silhouette. When the MBGS is started, the particles are evenly distributed and the system needed usually 20-50 frames to find a sufficiently good approximation of the true density. Then, the selected silhouette is rather random. After 50 frames, the silhouette with the maximum likelihood is the correct one in $\approx 98\%$ of the cases. In $\approx 20\%$ of the cases the

Fig. 2. BGS approach comparison of shadow issue

Fig. 3. BGS approach comparison of heavily occlusion

Fig. 4. BGS approach comparison of low contrast issue

ML silhouette was incorrect when e.g. a bush was largely occluding the legs. However, recovery time was within 5 frames. In case of partial occlusion of the entire body through, e.g. small trees, reliability degraded between 1% (slight occlusion) to 10% (considerable occlusion), The silhouette was incorrect in $\approx 59\%$ of the cases where the legs were fully occluded, e.g. by a car. In the videos the individual was in average 70 px. high. Reliability increased with more pixels on target.

The system has been tested on a 2 GHz Pentium under Linux. In videos of size 320×240 pixels with only a single person to track, the system runs, with 350 particles, with ≈ 50 ms/frame: ≈ 25ms/frame were used by the classical BGS, ≈ 25ms/frame were used by the matching.

5 Conclusion

The presented model-based background subtraction system combines the classical background subtraction with model knowledge of foreground objects. The application of model knowledge is not applied on a binary BGS image but on the "likelihood image", i.e. an image where each pixel value represents a confidence of belonging either to the foreground or background. A key issue in this study is the clustering of the silhouettes and the temporal diffusion step in the Bayesian propagation.

In the above application we have chosen silhouettes of humans, but we belive that this choice is without limit of generality since even different object types fit into the tree structure.

The presented experiments were carried out with only a single individual in the database. We have experimented also with different individuals (and thus varying silhouettes), but the output was instable w.f.t. the choice if the individual. This is under further investigation and the use of our approach for gait recognition is future research.

References

1. Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 142–149, Hilton Head Island, SC, June 13-15, 2000.
2. A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–209, 2000.
3. A. Elgammal and L. Davis. Probabilistic framework for segmenting people under occlusion. In *ICCV*, ICCV01, 2001.
4. D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *Proc. Int. Conf. on Computer Vision*, pages 87–93, Korfu, Greece, 1999.
5. I. Haritaoglu, D. Harwood, and L. Davis. W4s: A real-time system for detection and tracking people in 2.5 D. In *Proc. European Conf. on Computer Vision*, Freiburg, Germany, June 1-5, 1998.
6. T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proceedings of IEEE ICCV'99 FRAME-RATE Workshop*, 1999.
7. M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int. J. of Computer Vision*, 29:5–28, 1998.
8. Yuri A. Ivanov, Aaron F. Bobick, and John Liu. Fast lighting independent background subtraction. *Int. J. of Computer Vision*, 37(2):199–207, 2000.
9. A. Kale, A. Sundaresan, A.N. Rjagopalan, N. Cuntoor, A.R. Chowdhury, V. Krnger, and R. Chellappa. Identification of humans using gait. *IEEE Trans. Image Processing*, 9:1163–1173, 2004.
10. G. Kitagawa. Monta carlo filter and smoother for non-gaussian nonlinear state space models. *J. Computational and Graphical Statistics*, 5:1–25, 1996.
11. V. Krueger and S. Zhou. Exemplar-based face recognition from video. In *Proc. European Conf. on Computer Vision*, Copenhagen, Denmark, June 27-31, 2002.
12. J.S. Liu and R. Chen. Sequential monte carlo for dynamic systems. *Journal of the American Statistical Association*, 93:1031–1041, 1998.
13. K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 50–59, Vancouver, Canada, 9-12 July, 2001.

Estimation of Moments of Digitized Objects with Fuzzy Borders

Nataša Sladoje and Joakim Lindblad

Centre for Image Analysis,
Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden
{natasa, joakim}@cb.uu.se

Abstract. Error bounds for estimation of moments from a fuzzy representation of a shape are derived, and compared with estimations from a crisp representation. It is shown that a fuzzy membership function based on the pixel area coverage provides higher accuracy of the estimates, compared to binary Gauss digitization at the same spatial image resolution. Theoretical results are confirmed by a statistical study of disks and squares, where the moments of the shape, up to order two, are estimated from its fuzzy discrete representation. The errors of the estimates decrease both with increased size of a shape (spatial resolution) and increased membership resolution (number of available grey-levels).

Keywords: Fuzzy sets, moments, accuracy, multigrid resolution.

1 Introduction

Fuzziness is a natural outcome of most image processing techniques. It is an intrinsic quality of images and often provides important information about the image [1,5]. Fuzzy membership of a point reflects the level to which that point fulfils certain criteria to be a member of a set. The process of converting an input image to a fuzzy set by indicating, for each pixel, the degree of membership to the object, is referred to as *fuzzy segmentation*. Different imaging devices produce grey-level images with different underlying interpretations; a knowledge about such specific properties of the image is preferably incorporated into the fuzzy segmentation method that is applied. In practise, area coverage related memberships often result from the imaging process. Brightness of a pixel is proportional to the part of its area covered by the observed object, and grey-levels can therefore directly be used for defining membership values of a fuzzy object. Most image points are assigned memberships (close to) zero or (close to) one, so they can easily be classified as either object or background. Some of the points, often located around the border of an object, are assigned memberships between zero and one, according to the extent of their membership to the object. A segmentation resulting in a digital object with a fuzzy border is usually easier to perform than a crisp one.

In our previous work, it is shown that the precision of estimates of various properties of a continuous original shape is highly improved when derived from a

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 188–195, 2005.

fuzzy, instead of a crisp representation of a continuous object [4]. In this paper, we analyse the accuracy of the estimation of moments, when they are calculated from a fuzzy representation of a shape. We show that the order of magnitude of the error can be reduced if the estimation is based on a fuzzy discrete shape representation; a fuzzy approach can be an alternative to increasing the resolution of the image.

2 Moments Estimated from a Crisp Discrete Set

Definition 1. *The (p, q)-moment of a planar crisp set S in the Euclidean plane equipped with the Cartesian xy-coordinate system is defined by*

$$m_{p,q}(S) = \iint_S x^p y^q \, dxdy \, ,$$

for integers $p, q \geq 0$. The moment $m_{p,q}(S)$ has the order $p + q$.

If the set S is inscribed into an integer grid and digitized, instead of the moments $m_{p,q}(S)$, the moments of the digitization $\mathcal{D}(S)$ are available.

Definition 2. *The discrete moment $\tilde{m}_{p,q}(S)$ of a crisp set S is defined by*

$$\tilde{m}_{p,q}(S) = \sum_{(i,j)\in\mathcal{D}(S)} i^p j^q \, ,$$

where $\mathcal{D}(S)$ is the set of integer grid points contained in the set S.

The concept of moments is introduced in image analysis by Hu [2]. Several features of a shape can be calculated from a sufficient number of its moments. In fact, a shape can be recovered from an appropriately chosen set of its moments. If continuous moments are replaced by the their discrete counterparts, more or less good estimates of the observed features of a continuous shape can be obtained.

One of the main disadvantages of using moments for a description of a shape is that they are time consuming to compute; the problem increases with the increase of the image resolution. On the other hand, the accuracy of the estimations of the continuous shape features from the corresponding discrete moments increases with the increase of the image resolution. Consequently, it is desirable to use the information about the expected error of a moment estimation at a certain resolution, to get the best accuracy with the lowest possible computational effort, as discussed in [3,6]. In this paper, we study the possibility of using a fuzzy representation of a shape in order to increase the accuracy of moments estimation at a given spatial resolution. We use the results proved in [3], stated below, to derive error bounds for moments estimated from fuzzy sets.

Theorem 1. *The moments of a convex set S, with a boundary consisting of a finite number of C^3 arcs, digitized in a grid with resolution r (the number of grid points per unit), can be estimated from a set $rS = \{(rx, ry) \mid (x, y) \in S\}$ by*

$$m_{p,q}(S) = \frac{1}{r^{p+q+2}} \, \tilde{m}_{p,q}(rS) + \mathcal{O}\left(\frac{1}{r}\right), \tag{1}$$

for $p+q \leq 2$. If S is a 3-smooth convex set, its moments can be estimated by

$$m_{p,q}(S) = \frac{1}{r^{p+q+2}}\ \tilde{m}_{p,q}(rS) + \mathcal{O}\left(\frac{1}{r^{\frac{15}{11}-\varepsilon}}\right). \tag{2}$$

Remark 1: A planar 3-smooth convex set is a convex set in the Euclidean plane whose boundary consists of a finite number of arcs having continuous third order derivatives and a positive curvature at every point, except the end points of the arcs. These conditions exclude the existence of straight boundary segments.
Remark 2: The set $rS = \{(rx, ry) \mid (x, y) \in S\}$ is a dilation of the continuous set S by the grid resolution r.

3 Moments Estimated from a Fuzzy Discrete Set

Fuzzy sets are defined by Zadeh [7].

Definition 3. *A fuzzy subset S of a reference set X is a set of ordered pairs $S = \{(x, \mu_S(x)) \mid x \in X\}$, where $\mu_S : X \to [0, 1]$ is the membership function of S in X.*

We consider discrete fuzzy sets as representations of discretized continuous crisp shapes. Fuzzy moments and the centre of gravity of a fuzzy set are among the first defined fuzzy concepts.

Definition 4. *The discrete fuzzy moment $f\tilde{m}_{p,q}(S)$ of a fuzzy set S of a reference set $X \subset Z^2$ is*

$$f\tilde{m}_{p,q}(S) = \sum_{(i,j)\in X} \mu_S(i,j)\ i^p j^q\ ,$$

where $\mu_S(i, j)$ is the membership function of S.

In the following, $c\tilde{m}_{p,q}(S)$ denotes a crisp discrete moment of a set S.

We analyze and compare the error bounds for the two approaches, i.e., when crisp and fuzzy discrete moments, respectively, are used in the estimation of the moments of a continuous shape.

The question we would like to answer is: *If we are not able to increase the spatial resolution of an image in order to achieve an increased accuracy of the shape moment estimation, how much do we gain if we use a fuzzy representation of the object instead of a crisp one?*

To answer this question, we observe a pixel P_f in a grid with a spatial resolution r_s and membership resolution r_f^2 (the number of grey-levels per unit, here equal to the number of grey-levels available), and compare it with a block P_c of $r_f \times r_f$ pixels in a grid of a spatial resolution $r_s r_f$ and membership resolution 1. Fuzzy membership of a pixel P_f is assigned according to its area coverage, and is approximated by k/r_f^2. This corresponds to the number of sub-pixels, denoted by k (out of r_f^2), within the pixel P_f, which are covered by the observed continuous

crisp shape S. For a sub-sampling factor r_f, the number of possible grey-levels is r_f^2, which requires a pixel depth of $2\log_2(r_f)$ for representation. In the alternative approach, we consider an r_f times dilated crisp representation of a shape. For the dilated shape, the pixels within $P_c \cap S$ correspond to the sub-pixels within P_f, covered by S. Their number is equal to k, for $k \in \{0, 1, \ldots, r_f^2\}$.

We assume that the size of a pixel is equal to 1. Consequently, the size of P_c is equal to $r_f \times r_f$. The coordinates of a pixel P_f are $(r_s x, r_s y)$, while the coordinates of the pixels within a block P_c can be expressed as

$$\left(r_s r_f x - \frac{r_f}{2} + \frac{1}{2} + i,\ r_s r_f y - \frac{r_f}{2} + \frac{1}{2} + j \right),$$

where $i, j \in \{0, 1, \ldots, r-1\}$.

3.1 Zero-Order Moment Estimation

The contribution of the pixel P_f to the fuzzy discrete moment $f\tilde{m}_{0,0}(r_s S)$ is

$$f\tilde{m}_{0,0}^{(1)}(r_s S) = \frac{k}{r_f^2}.$$

The upper-index (1) denotes the contribution of one fuzzy pixel and one block of crisp pixels, respectively.

Alternatively, if the grid resolution r_s is increased r_f times, there are k, out of r_f^2, pixels having their centroids within the continuous crisp shape and contributing to the zero-order crisp moment. The contribution of P_c to the moment $c\tilde{m}_{0,0}(r_s r_f S)$ is

$$c\tilde{m}_{0,0}^{(1)}(r_s r_f S) = k.$$

Considering all $r_s \times r_s$ fuzzy pixels/crisp blocks in the image, we derive the following relation between the crisp and the fuzzy discrete moments of the observed (discrete) shape:

$$c\tilde{m}_{0,0}(r_s r_f S) = r_f^2\ f\tilde{m}_{0,0}(r_s S). \tag{3}$$

By using (3) and Theorem 1, it follows that the zero-order moment of a continuous convex shape S can be estimated by

$$m_{0,0}(S) = \frac{1}{r_s^2 r_f^2}\ c\tilde{m}_{0,0}(r_s r_f S) + \mathcal{O}\left(\frac{1}{r_s r_f}\right) = \frac{1}{r_s^2}\ f\tilde{m}_{0,0}(r_s S) + \mathcal{O}\left(\frac{1}{r_s r_f}\right), \tag{4}$$

while the accuracy of the estimation of the $m_{0,0}(S)$ moment of a 3-smooth convex shape S can be expressed by

$$m_{0,0}(S) = \frac{1}{r_s^2}\ f\tilde{m}_{p,q}(r_s S) + \mathcal{O}\left(\frac{1}{(r_s r_f)^{\frac{15}{11}-\varepsilon}}\right). \tag{5}$$

Thus, we conclude that using r_f^2 pixel membership values provides the same accuracy of $m_{0,0}(S)$ estimation as increasing the (crisp) image spatial resolution r_f times.

3.2 First-Order Moments Estimation

Moments of order higher than zero are position variant. At low resolutions, the accuracy of the estimation cannot be fully preserved if the spatial resolution is r_f times decreased, and fuzzy resolution r_f^2 is used instead. However, the asymptotic expressions that we derive show that such a behaviour is expected if the spatial resolution is high enough.

The contribution of the observed pixel P_f (having coordinates $(r_s x, r_s y)$) to the moment $f\tilde{m}_{1,0}(r_s S)$ is

$$f\tilde{m}^{(1)}_{1,0}(r_s S) = r_s x \, \frac{k}{r_f^2}.$$

Note that only the number of covered sub-pixels is taken into account, and not their coordinates.

On the other hand, if the grid resolution r_s is increased r_f times, there are k out of r_f^2 pixels having their centroids within the continuous crisp shape and therefore contributing to the first-order crisp moment. The contribution of P_c to the moment $c\tilde{m}_{1,0}(r_s r_f S)$ is

$$c\tilde{m}^{(1)}_{1,0}(r_s r_f S) = \left(r_s r_f x - \frac{r_f}{2} + \frac{1}{2} \right) k + \sum i,$$

where i takes k values from the set $\{0, 1, \ldots, r-1\}$, depending on the position of the k covered pixels within P_c.

It is easy to conclude that in the case when $k = r_f^2$, i.e., when all pixels within the block P_c are covered by S, and, equivalently, the membership value of P_f to the observed shape is equal to 1,

$$c\tilde{m}^{(1)}_{1,0}(r_s r_f S) = r_f^3 \; f\tilde{m}^{(1)}_{1,0}(r_s S). \tag{6}$$

Under the assumption made about the fuzzy representation of the observed convex shape, $\mathcal{O}\left(r_s^2\right)$ of the image points (all of the inner pixels) are of this type.

The pixels on the border of a shape have membership values between 0 and 1. The biggest difference between the contributions of P_f to the fuzzy moment and of the block P_c to the crisp moment appears when one half of a pixel/block (left or right part) is covered:

$$c\tilde{m}^{(1)}_{1,0}(r_s r_f S) = r_f^3 \; f\tilde{m}^{(1)}_{1,0}(r_s S) + \mathcal{O}\left(r_f^3\right). \tag{7}$$

We assume that there are $\mathcal{O}\left(r_s\right)$ pixels of this type in the observed fuzzy image.

Considering (6) and (7), we derive the following relation between the crisp and the fuzzy discrete moments of an observed (discrete) shape:

$$c\tilde{m}_{1,0}(r_s r_f S) = r_f^3 \; f\tilde{m}_{1,0}(r_s S) + \mathcal{O}\left(r_s r_f^3\right). \tag{8}$$

By using (8) and Theorem 1, it follows that the first-order moment of a continuous convex shape S can be estimated by

$$m_{1,0}(S) = \frac{1}{r_s^3} \; f\tilde{m}_{1,0}(r_s S) + \mathcal{O}\left(\frac{1}{r_s^2}\right) + \mathcal{O}\left(\frac{1}{r_s r_f}\right). \tag{9}$$

We conclude that once the spatial resolution is high enough to fully "exploit" the fuzzy membership values of pixels, i.e., when $r_s > Cr_f$, where C is a constant derived from the asymptotic expression for the error bound, using r_f^2 pixel membership values provides the same accuracy of $m_{1,0}(S)$ estimation as increasing the (crisp) image spatial resolution r_f times.

By using (8) and the second part of Theorem 1, for a 3-smooth convex shape S it holds

$$m_{1,0}(S) = \frac{1}{r_s^3}\, f\tilde{m}_{1,0}(r_s S) + \mathcal{O}\left(\frac{1}{(r_s r_f)^{\frac{15}{11}-\varepsilon}}\right), \qquad \text{for } r_s > C r_f{}^{\frac{15}{7}+\varepsilon}. \tag{10}$$

Analogous results follow for the $m_{0,1}(S)$ moment estimation.

3.3 Second-Order Moments Estimation

In the derivation of the asymptotic expressions for the estimation of the second-order moments of a convex shape from its fuzzy representation, we apply similar reasoning as for the first-order moments.

For $\mathcal{O}\left(r_s^2\right)$ fully covered pixels P_f/blocks P_c, it holds that

$$c\tilde{m}_{2,0}^{(1)}(r_s r_f S) = r_f^4\; f\tilde{m}_{2,0}^{(1)}(r_s S) + \mathcal{O}\left(r_f^4\right), \tag{11}$$

while for $\mathcal{O}\left(r_s\right)$ pixels/blocks on the border of the object, in the worst case (if the right half of the pixel/block is covered by the object) it holds that

$$c\tilde{m}_{2,0}^{(1)}(r_s r_f S) = r_f^4\; f\tilde{m}_{2,0}^{(1)}(r_s S) + \mathcal{O}\left(r_s r_f^4\right). \tag{12}$$

Considering (11) and (12), for the whole image it follows that

$$c\tilde{m}_{2,0}(r_s r_f S) = r_f^4\; f\tilde{m}_{2,0}(r_s S) + \mathcal{O}\left(r_s^2 r_f^4\right). \tag{13}$$

By using (13) and Theorem 1, it follows that the second-order moment of a continuous convex shape S can be estimated by

$$m_{2,0}(S) = \frac{1}{r_s^4}\, f\tilde{m}_{2,0}(r_s S) + \mathcal{O}\left(\frac{1}{r_s r_f}\right), \quad \text{for } r_s > C r_f. \tag{14}$$

As in the case of the first order moment estimation, we conclude that using r_f^2 pixel membership values asymptotically provides the same accuracy of $m_{2,0}(S)$ estimation as increasing the spatial resolution of the (crisp) image r_f times.

Similarly, from (13) and the second part of Theorem 1, it follows

$$m_{2,0}(S) = \frac{1}{r_s^4}\, f\tilde{m}_{2,0}(r_s S) + \mathcal{O}\left(\frac{1}{(r_s r_f)^{\frac{15}{11}-\varepsilon}}\right), \quad \text{for } r_s > C r_f{}^{\frac{15}{7}+\varepsilon}. \tag{15}$$

Analogous results follow for the $m_{1,1}(S)$ and $m_{0,2}(S)$ moments estimation.

4 Statistical Study of Squares and Disks

We perform a statistical study in order to examine the properties of moments estimated at low resolutions. Multigrid resolution is expressed by dilations of the observed objects. Tests are performed for $2,000$ randomly positioned disks (the centres are randomly positioned inside a pixel) for each of a number of observed real-valued radii within the interval $[1, 100]$, and for $10,000$ randomly positioned squares for each observed real-valued side length within the interval $[1, 90]$ (for each size, 100 random centre positions, each with 100 random rotations between 0 and 45 degrees, are considered). Different membership resolutions are used ($r_f \in \{1, 2, 4, 8, 16\}$). Note that the $r_f = 1$ corresponds to crisp segmentation, and that $r_f = 16$ gives the upper limit for the membership resolution of 8-bit pixel depth.

For each size of an object, we determine the maximal relative estimation error for moments up to the order two. We present the results for $m_{1,0}$ and $m_{2,0}$ moments estimation, both for squares and for disks, in Figure 1. The estimation

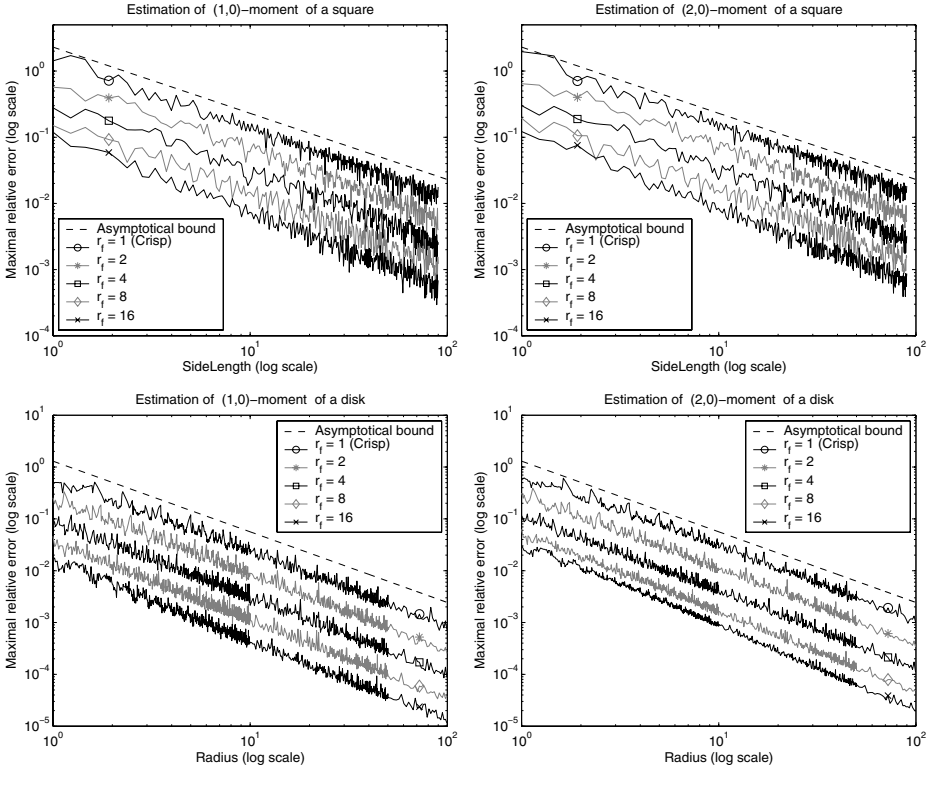


Fig. 1. Error bounds for first and second order moments estimation. *Top:* Moments estimation of a square. *Bottom:* Moments estimation of a disk.

errors for squares show asymptotic behaviour in accordance with expressions (8) and (13). Disks are 3-smooth convex objects, and the corresponding estimation bounds agree with expressions (10) and (15).

Plots are presented in a logarithmic scale so that the "slopes" of the curves correspond to the order of estimation error, and can be compared with the plotted straight line which has a slope equal to the theoretically derived order of error (-1 for squares and $-\frac{15}{11}$ for disks). The expected asymptotic behaviour assumes higher errors at low resolutions, which is related to the value of the constant C in (10), (14), and (15). However, for the presented membership resolutions, the plots show accordance with the asymptotic bounds also at low spatial resolutions. The relative position of the curves shows that the estimation error becomes smaller both with the increase of spatial and membership resolution.

5 Comments and Conclusions

Derived asymptotic expressions for the estimation of moments of convex shapes from their fuzzy discrete representation show that an increase in membership resolution of an image can be used to achieve the same accuracy of the estimation as an increase of the spatial resolution. Even though the theoretical results guarantee this behaviour only after a certain spatial resolution is reached, the simulations show that it is also present at low resolutions.

By using a fuzzy, instead of a crisp, representation of a shape, which in many cases is easily obtained from the imaging device, significant improvements of the accuracy of moments estimation (and shape features derived from them) are achievable. By fully utilizing an often already existing membership resolution, it is possible to overcome problems of insufficient available spatial resolution.

Acknowledgements

Prof. Gunilla Borgefors and Doc. Ingela Nyström, both Centre for Image Analysis, Uppsala, Sweden, are gratefully acknowledged for their scientific support.

References

1. I. Bloch and H. Maître. Fuzzy mathematical morphologies: A comparative study. *Pattern Recognition*, 28(9):1341–1387, 1995.
2. M. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Inform. Theory*, 8:179–187, 1962.
3. R. Klette and J. Žunić. Multigrid convergence of calculated features in image analysis. *Journal of Mathematical Imaging and Vision*, 13:173–191, 2000.
4. N. Sladoje, I. Nyström, and P. Saha. Measurements of digitized objects with fuzzy borders in 2D and 3D. *Image and Vision Computing*, 23:123–132, 2005.
5. J. K. Udupa and G. J. Grevera. Go digital, go fuzzy. *Pattern Recognition Letters*, 23:743–754, 2002.
6. J. Žunić and N. Sladoje. Efficiency of characterizing ellipses and ellipsoides by discrete moments. *IEEE Trans. on PAMI*, 22(4):407–414, 2000.
7. L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

Feature Matching and Pose Estimation Using Newton Iteration

Hongdong Li and Richard Hartley

Research School of Information Sciences and Engineering,
The Australian National University,
ASSeT, Canberra Research Labs, National ICT Australia

Abstract. Feature matching and pose estimation are two crucial tasks in computer vision. The widely adopted scheme is first find the correct matches then estimate the transformation parameters. Unfortunately, such simple scheme does not work well sometimes, because these two tasks of matching and estimation are mutually interlocked. This paper proposes a new method that is able to estimate the transformation and find the correct matches simultaneously. The above interlock is disentangled by an alternating Newton iteration method. We formulate the problem as a nearest-matrix problem, and provide a different numerical technique. Experiments on both synthetic and real images gave good results. Fast global convergence was obtained without the need of good initial guess.

1 Introduction

In many vision applications, one often encounters the problem of estimating the relative geometric transformation (i.e.,pose) between two feature point sets in two images. The most straightforward and widely-adopted way of doing it is first establish point-wise feature matches (correspondences) between the two feature sets and then estimate the transformation parameters. However, finding good correspondences is not easy, especially when two images are linked by some geometric transformations with unknown parameters.

Matching and estimation are two mutually interlocked problems: finding one is often the precondition for finding the other. On one hand, if the transformation is roughly known, then matching becomes easier. The local-correlation-based matching method will generally not work if no information about the transformation is given; on the other hand the transformation parameter can only be precisely estimated when correct matches are given. Thus matching and estimation form a chicken-and-egg dilemma.

This paper proposes a *global* method that is able to estimate the transformation parameters and find correct matches simultaneously. We formulate the matching-and-estimation problem as a *nearest matrix problem*, in particular to find an orthogonal *polar factor* matrix [8]. This compact formulation allows for numerically stable methods. We suggest a very neat *Newton root-finding method*, (the well known *tangent-line method*), which is well understood and has good

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 196–203, 2005.

theoretical properties. Experimental results on both synthetic feature sets and real imagery gave encouraging results.

2 Matching as a Nearest Matrix Problem

2.1 Global Matching Methods

The majority of existing matching methods relies on local similarity of the feature points. One example is the correlation technique. These methods are well understood and widely adopted. However, the main disadvantages of these methods are that they are very sensitive to noise, and some crucial parameters (such as the size of neighborhood and correlation threshold) must be empirically chosen. Researchers have noted that global properties such as rigidity and exclusiveness are very important for matching. A number of global methods have been proposed. Among them, the spectral graph theory methods [5,3,4,9], global cost minimization methods [2,6], the epipolar constraint method [12], are some nice examples. Such global matching methods seem to be more promising than local methods. We propose a global matching method in this paper. By *global*, we mean that the best correspondences are the ones that minimize a global cost function. Encoded in this cost function are many global properties within and between the two feature sets. The matching problem is cast as a *nearest matrix approximation* problem. The special form of this nearest matrix encapsulates the global properties.

2.2 Nearest Matrix Problem

Suppose we are given two discrete feature sets in two images. For the moment let us assume that these two sets have an identical size of N, and that all feature points have matches. This restriction will be relaxed later.

We assign to each feature an attribute vector, which could be its pixel coordinates $[u, v]$ and/or the gray values within a local neighborhood $[g1, g2, \ldots]$. Let it be `row vector` of dimension **d**. Stacking all the vectors of each image together, we thus obtain two *feature matrices*, X and Y, of size **Nxd**, which describe the feature points in image 1 and image 2, respectively. Let us, for the moment, further assume that there is no significant geometric transformation between two feature sets, except for some jitters in coordinates. We use a permutation matrix P of size **NxN** to express the feature matching relations. A permutation matrix is a square matrix whose entries are either zero or one, and the entries of each row and of each column add up to one.

In the matching problem, the positions of one elements in P describe the correct matching between X and Y, that is, $\mathrm{X} \leftrightarrow \mathrm{PY}$. So, when all the correct matches are found, the following function J should reach its minimum:

$$J = \| \mathrm{X} - \mathrm{PY} \|_F^2 \rightarrow min \tag{1}$$

This J is the global cost function for the matching problem. The particular P corresponding to the minimum value of J gives the correct matches. In fact,

such J is a distance measurement describing the proximity of two feature sets, and is known as the *standard Procrustes distance* [11]. From (1) and by some linear algebra manipulations, we have

$$\begin{aligned} min \parallel \mathrm{X} - \mathrm{PY} \parallel_F^2 &\Leftrightarrow minTr((\mathrm{X} - \mathrm{PY})^T(\mathrm{X} - \mathrm{PY})) \\ &\Leftrightarrow Tr(\mathrm{X}^T\mathrm{PY}) \\ &\Leftrightarrow Tr(\mathrm{PYX}^T) \\ &\Leftrightarrow min \parallel \mathrm{P} - \mathrm{XY}^T \parallel_F^2 \end{aligned} \tag{2}$$

Defining $\mathrm{A} = \mathrm{XY}^T$, this problem becomes one of finding a real permutation matrix P such that

$$\mathrm{P} = argmin \parallel \mathrm{P} - \mathrm{A} \parallel_{\mathrm{F}}^2 \tag{3}$$

This is a typical nearest matrix problem. The formulation is more compact than (1) and has many benefits which will be explained later. Now, we reach the following result: *Given matrix* A *defined as above, find a permutation matrix* P *which is nearest to* A. *Then this* P *gives the best correspondences.*

Many methods use the doubly stochastic ($\mathcal{DS}$) property of the permutation matrix. By contrast, this paper places more emphasis on the orthogonality of P. That is, a permutation matrix must be an orthogonal matrix: $\mathrm{PP}^T = \mathrm{I}_n$.

Similar idea has been used by the spectral matching approaches such as [4] [9], but we find the orthogonality property is so strong that an iteration algorithm can be derived directly from it. This results in a very neat Newton root-finding method. Specifically, the best P is no other than the nearest (with respect to A) square root matrix of the identity matrix I. Moreover, according to our experiments, we found that the resultant matrix P is close enough to the ideal zero-one permutation, even when other constraints of a permutation matrix have not been imposed.

3 Matching and Estimation at the Same Time

When there are geometric transformations between the two feature sets, formula (1) no longer hold. However, we can easily modify it by accounting such transformation. We confine the transformation to be linear only. This is a limitation of our method, but not too serious, because it covers most commonly encountered transformations, such as Euclidean motion, or affine transformation. Denote the linear transformation by a matrix R, Equation (1) then becomes

$$J = \parallel \mathrm{X} - \mathrm{PYT} \parallel_F^2 \rightarrow min \tag{4}$$

where $\mathrm{T}_{d\times d} = \begin{bmatrix} \mathrm{R} & 0 \\ 0 & \mathrm{I}_{d-2} \end{bmatrix}$. The sub-identity matrix I in T is used for those non-coordinate attribute components of X and Y. Following similar derivations as in (2), equation (3) is modified to :

$$\mathrm{P}_{n\times n} = argmin \parallel \mathrm{P} - \mathrm{XT}^T\mathrm{Y}^T \parallel_F^2 \tag{5}$$

If the transformation T is known in advance, the matching problem can be solved by finding the nearest permutation matrix to the matrix P. On the other hand, if we already know the correspondences P, then the transformation T can be estimated by solving another nearest matrix problem: $\mathrm{T}_{d\times d} = argmin \parallel \mathrm{T} - \mathrm{Y}^T\mathrm{P}^T\mathrm{X} \parallel_F^2$, where T is subject to some special constraints that correspond to the particular transformation model. For example, in the case of a 2d rotation, R should be an orthogonal matrix with unit positive determinant. For the sake of computational simplicity, T can also be approximated as $\mathrm{T} \approx \mathrm{Y}^T\mathrm{P}^T\mathrm{X}$ or $\mathrm{T} \approx \mathrm{Y}^\dagger\mathrm{P}^T(\mathrm{X}^T)^\dagger$, where † denotes the pseudo-inverse. In our problem, both the transformation and correspondences are not known beforehand. In order to solve this dilemma, we use an alternating iteration approach.

4 The Proposed Newton Iteration Algorithm

The structure of the alternating iteration algorithm for simultaneously solving P and T is given as follows:

Begin
Initialize T_0.
For k = 0,1,2,3,..., do the following until convergence:

$$\mathrm{P}_{k+1} = argmin \parallel \mathrm{P}_k - \mathrm{X}^T\mathrm{T}_k^T\mathrm{Y} \parallel_F^2 \tag{6}$$

$$\mathrm{T}_{k+1} = argmin \parallel \mathrm{T}_k - \mathrm{Y}^T\mathrm{P}_k^T\mathrm{X} \parallel_F^2 \tag{7}$$

End.

At each iteration step, both P and T should also be adjusted to comply with certain constraints. Since (6) and (7) generate a non-increasing sequence, convergence is ensured. There is actually no preference as to which one is performed before the other, however, by considering our application, the number of feature points is often greater than the number of parameters of T. Thus it is more reasonable to make an initial guess of T and first solve for P.

Equation (7) can be directly solved as a minimization problem, as in [6]. Alternatively, we can simply apply the approximations of the last section. The latter method is simpler and more effective. In our implementation, the orthogonal condition is realized by a matrix version Newton root-finding algorithm. Besides the nearest matrix formulation, this Newton matrix algorithm is another distinct feature of our method.

4.1 Matrix Root Problem and Matrix Newton Method

Consider the problem of finding the real roots of a nonlinear scalar equation $f(x) := x^2 - 1 = 0$. Newton's tangent line method, $x_{k+1} = x_k + f(x_k)/f'(x_k)$, gives the iteration formula $x_{k+1} = (x_k + x_k^{-1})/2$.

In our problem, P is a square root matrix of the identity matrix I. To solve such *matrix root problem*, Higham [8] proposed a numerical iteration method by mapping the original scalar form Newton root-finding method to a matrix form. Analogously, we obtain $\mathrm{P}_{k+1} = (\mathrm{P}_k + \mathrm{P}_k^{-T})/2$.

Of course, for rigorous theoretic justification of this formula, the reader is referred to [8], which has also proven its quadratic convergence. Now the minimization of (6) is realized by the above Newton iteration with $\mathrm{P}_0 = \mathbf{exp}(\mathrm{X}^T\mathrm{T}^T\mathrm{Y})$ as the starting point. Here the function **exp** can be thought of a nonlinear kernel function, one of which purposes is to ensure the positivity of the resulting entries. As it may be a singular matrix we actually used pseudo-inverse instead. Experiments showed that the final P was very likely to have entries that were very close to zero or one. Notice that we have not enforced any doubly stochastic constraints upon P.

Similarly, when the transformation is simply a 2D Euclidean motion, e.g, a rotation matrix R, it can also be solved by such Newton iteration, namely,$\mathrm{R}_{k+1} = (\mathrm{R}_k + \mathrm{R}_k^{-T})/2$. Besides, we enforce a constraint $\mathbf{det}(\mathrm{R}) = +1$ to make it a proper rotation matrix. For this purpose, we use a simple trick by swapping any two rows of R when needed. Our experiments showed that by this constraint we are able to reach the global optimum even from arbitrary initial guesses.

4.2 Some Practical Considerations

Compare our algorithm with the annealing-based softAssign or SoftPOSIT method. If we roughly think of the orthogonal matrix P_k as a high-dimensional rotation, then the effect of each iteration is to gradually rotate the correspondences towards a consistent configuration. On the other hand, the doubly stochastic annealing strategy consists of first preferring a uniformly distribution of correspondence credits, and then choosing one winner that has the most response according to a winner-take-all consensus.

In practice, we often meet cases where the sizes of the two feature sets are not equal, or where some feature points are occluded or fail detected (outliers). These cases will not degrade our method, because to deal with them we need only make small modifications of the proposed algorithm.

When set sizes are unequal, matrix P is no longer square and no longer an orthogonal matrix, but it is still an orthonormal matrix. Therefore Newton method is still applicable. When there are outliers, we can augment P with one row and one column. The newly added row or column corresponds to a so-called *slack feature* that is able to match any point.

5 Experiments and Results

We carried out several experiments on both synthetic data and real images. The first experiment tested the matching performance with noisy data. We generated a discrete point set with $N = 79$ feature points along the contour of a pine tree shape. This is taken as the model set. The other sets were generated by randomly

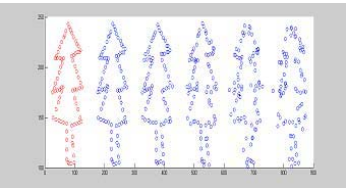


Fig. 1. Pine model set, and data sets with various noise levels,(*sigma*= 0.5, 1.0, 2.0, 3.0, 4.0)

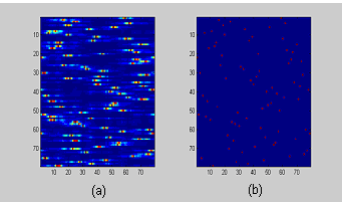


Fig. 2. The correlation matrix A before (left), and after Newton iteration (right)

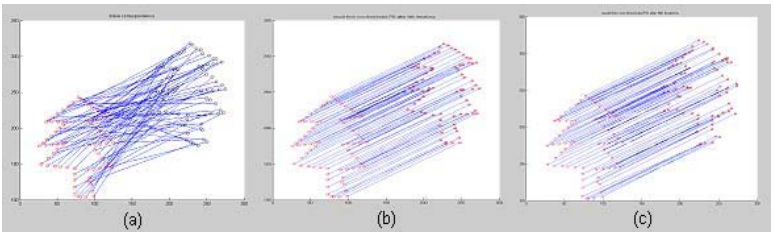


Fig. 3. Feature matching results (a) random initial correspondences; Results (b)*sigma* = 1.0 (c) *sigma* =4.0

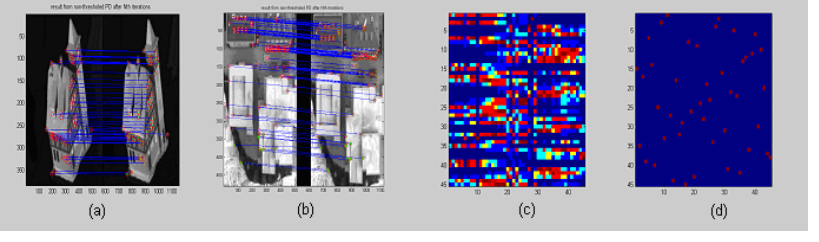


Fig. 4. Matching results on real images

perturbing the coordinates of the model set. Both are shown in figure-1. These perturbations were drawn from a zero-mean isotropic Gaussian distribution with standard deviation sigma in the range [0.5 4.0 pixels]. Note that in particular the perturbations are rather large with respect to the minimum distance between model points (only about 3-4 pixels). In other words, the perturbations caused large shape distortions in the dense model features. This is a challenging situation for local correlation based matching algorithms.

Figure-2 (left) shows (in pseudo-color graph) an initial matrix A corresponding to a random initial correspondence. Matrix A is the input to our Newton algorithm. The resultant nearest permutation matrix P is depicted in figure(2)-(right). Using this P, we obtained the matching results shown in figure-3. All the results are fairly good. No serious bad matches were found. Figures-4 (a) and (b)

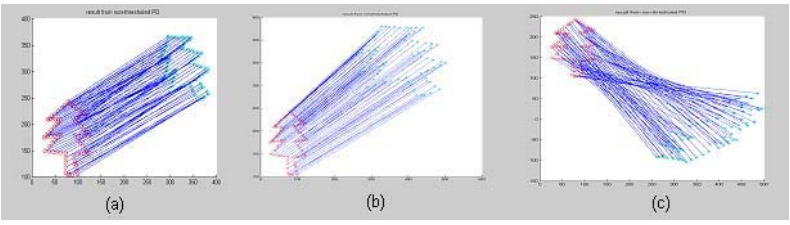


Fig. 5. Simultaneous matching and transformation estimation results; (a) rotation: real 39.0 degree, estimated 39.01 degree; (b) affine (only test 4-parameters): real [2.2, -0.8; 0.6, 1.1], estimated [2.1988 -0.7999; 0.6001 1.0998]; (c) affine (only test 4-parameters): real [+2.2, -0.8; +0.6, -1.1], estimated [2.2012, -0.7993; 0.5978, -1.0996]

show two test results on real images. Feature points were detected using a Harris corner detector. A subset was then manually selected. The attribute vector of each point consisted of its coordinates and the gray values of a 3x3 neighborhood. Also note that the feature number is rather large (Building: 121 points; House: 45 points), and are densely located. Remember that we actually used two different images that were drawn from a video sequence. Therefore their viewpoints and feature point coordinates are all different. We consider these differences as noise, and note that the matching results are still very satisfactory and encouraging. Figure 4(c) illustrates the initial values of entries of matrix A for the house images, and figure-4(d) shows the resulting matrix A after the Newton iteration. It is seen that this matrix has converged to a binary 0-1 matrix. In the second set of experiments, we tested the performance of simultaneous matching and transformation estimation. Figure 5 gives the matching and transformation estimation result on the synthetic graphs. For all the experiments conducted in figure 5, we did not make any initial guess for the transformation, except for knowing their models (Euclidean or affine). Our method has shown to be more robust, in the sense of global convergence, it seems less likely to be trapped in a local minima. In all experiments, the convergence were rather fast. Actually, the total convergence rate of our algorithm is no longer quadratic, but linear because of the alternative operation.

6 Discussion and Conclusions

Feature matching is no more than finding a permutation matrix. Many existing algorithms (such as the SoftAssign method and SoftPOSIT method) implement this by finding a doubly-stochastic matrix as an approximation. For NxN real matrix, there are N! possible choices. In order to find the one that best explains the input data, they often adopt an annealing procedure. The major difficulty is that the convergence is relatively slow and yet may get trapped in local minima. Notice that a doubly stochastic matrix in general is not an orthogonal matrix. Rather than using the doubly stochastic property in each search step, this paper uses a much stronger condition of the orthogonal property.

The proposed algorithm can be thought of *projecting* a matrix into the nearest *orthogonal manifold*. Because for NxN real matrix there are at most N orthogonal matrices, in turn, the chance to quickly find the optimal solution increases. This idea was used by many spectral graph matching algorithms[4][5][12], where they apply the SVD factorization. But here we suggest using Newton root-finding method, which has better numerical properties, and is essentially equivalent to finding the matrix **polar factor** only ([9]), therefore saves many computations. In other words, a full SVD decomposition is no need here.

Our method is a *global* matching method because many global Gestalt factors, such as proximity, similarity and exclusiveness, have been encoded in the nearest matrix formulation in a concise way. For cases where mismatching points exist, we consider using .EM-like algorithm to resolve it. By applying a specific parametric model this method could be easily extended to the matching of non-rigid deformation objects. These two topics will be investigated in the future.

Acknowledgements. National ICT Australia is funded through the Australian Government's Backing_Australia's_Ability_Initiative, in part through the Australian Research Council. The real images were taken from CMU-RI website. The authors are grateful to the anonymous reviewers for very valuable suggestions.

References

1. P.David, D. Dementhon, R.Duraiswami, H.Samet, SoftPOSIT: Simultaneous Pose and correspondence determination, ECCV-2002. Denmark. 2002.
2. S.Gold, S.Pappu, C.Lu, A.Rangarajan, E.Maolsness, New algorithm for 2D and 3D point matching: Pose estimation and correspondence, PR(31), pp1019-1031, 1998.
3. Scott, G.L., and Longuet-Higgins, H.C., An Algorithm for Associating the Features of Two Images, Proc. Royal, Soci. London B-244, 1991, pp. 21-26.
4. Shapiro, L.S., and Brady, J.M., Feature-Based Correspondence: An Eigenvector Approach, IVC (10), No. 5, June 1992, pp. 283-288.
5. S.Scalaroff, A. Pentland, Modal Matching for correspondence and recognition, T-PAMI 17(6), 1995, pp545-561.
6. Maciel and Costeira , A global solution to sparse correspondence problems, T-PAMI, 25-2, 2003
7. J.Kosowsky,a.Yuille, The invisible hand algorithm: solving the assignment problem with statistical physics, Neural networks, vol-7,1994, pp477-490.
8. N. Higham, Stable iteration for the matrix square root, Numerical algorithm 15, 1997, pp 227-242.
9. Carcassoni, M, Hancock, E., Spectral correspondence for point pattern matching, PR(36), No. 1, pp. 193-204. January 2003,
10. Anand Rangarajan, Haili Chui and Fred L. Bookstein, The Softassign Procrustes Matching Algorithm, James Duncan and Gene Gindi, editors , Information Processing in Medical Imaging, 1997.
11. D.Kendall.Shape Manifolds, Procrustean metrics and complex projective spaces, Bullet. London.Math.Society, 16:81-121,1984.
12. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd edition, Cambridge University Press, 2004.

Uncertainty Analysis of Camera Parameters Computed with a 3D Pattern

Carlos Ricolfe-Viala and Antonio-José Sánchez-Salmerón

Department of Systems Engineering and Automatic Control
Polytechnic, University of Valencia,Valencia, Spain
{cricolfe, asanchez@isa.upv.es}, http://www.isa.upv.es

Abstract. Camera calibration is a necessary step in 3D modeling in order to extract metric information from images. Computed camera parameters are used in a lot of computer vision applications which involves geometric computation. These applications use camera parameters to estimate the 3D position of a feature in the image. Depending on the accuracy of the computed camera parameter, the precision of the position of the image feature in the 3D scene vary. Moreover if previously the accuracy of camera parameters is known, one technique or another can be choose in order to improve the position of the feature in the 3D scene.

Calibration process consists of a closed form solution followed by a non linear refinement. This non linear refinement gives always the best solution for a given data. For sure this solution is false since input data is corrupted with noise. Then it is more interesting to obtain an interval in which camera parameters are contained more than an accurate solution which is always false.

The aim of this paper is to present a method to compute the interval in which the camera parameter is included. Computation of this interval is based on the residual error of the optimization technique. It is know that calibration process consists of minimize an index. With the residual error of the index minimization an interval can be computed in which camera parameter is. This interval can be used as a measurement of accuracy of the calibration process.

Keywords: camera calibration, accuracy evaluation, interval estimation, 3D pattern.

1 Introduction

Calibration of cameras is considered as an important issue in computer vision. It allows extracting metric information from 2D images. Accurate calibration of cameras is especially crucial for applications that involve quantitative measurements such as dimensional measurements. Calibration process computes camera parameters which are divided into internal and external. Internal camera parameters determine how the image coordinates of a point is derived, given the spatial position of the point with respect to he camera. Internal parameters are also those which define the quality of the image. They measure the camera distortions. These parameters are non linear related with the data input coordinates. By the other hand, position of the camera with

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 204 – 211, 2005.

respect to the scene is defined with the external parameters. In this case a linear relation exists.

Existing camera calibration techniques can be divided into several categories. From the point of view of the calibration object they use, they can be classified into four categories. First uses 3D reference calibration object. In this case, calibration object consist of two or three planes orthogonally to each other. This requires an elaborated calibration apparatus but calibration can be done very efficiently [2]. Second, calibration process is based of the observation of a planar pattern shown at different orientations [11]. Any knowledge about camera motion is necessary. The calibration setup is easier but less accurate results are computed. Third 1D calibration object is used. In this case very easy calibration setup is necessary [12]. However accuracy of the computed parameters is worst. The fourth group is called self-calibration. These calibration techniques are those which do not use any calibration object because only image point correspondences are required. Just by moving a camera in a static scene, the rigidity of the scene provides in general two constraints [7] on the cameras' internal parameters. If the images are taken by the same camera with fixed internal parameters, correspondences between three images are sufficient to recover both internal and external parameters.

All calibration techniques give a closed form solution for the camera parameters. Afterwards, an iterative scheme is used improve the results and to estimate parameters with non linear relation. They are radial distortion and geometrical misalignments. Taking into account that a non linear iterative scheme is done, final camera parameters are similar if one calibration object or another is used. They just give an initial value of the parameters to initialize non linear estimation and the iterative scheme give "the best" solution. Moreover, the computed parameter will never be the right ones since the input data is computed with noise. They will be the best which fit given data. From this point of view it is more interesting to obtain an interval in which the parameter is contained than computed an optimal solution for given data which will be wrong for sure.

In this paper a method to compute a interval in which a camera parameter is contained is presented. It is based on the residual error coming from the minimization algorithm. Depending on the minimization algorithm used to calibrate the camera the interval will be bigger or smaller. In this case the process is particularized to the calibration process using 3D calibration object since it gives more accurate solution using a closed form solution [1],[8]. One advantage of this calibration method is that any inverse matrix is computed. That means, from a mathematical point of view that, this method is robust and good performance is achieved with no well conditioned matrixes. First a brief presentation of the calibration process is done. Second, computing interval method is presented following with the results of both simulated and real experiments.

2 Camera Calibration Using a 3D Pattern

If $X_i=(x_i, y_i, z_i, 1)^T$ represents a point of the 3D calibration object and $U_i=(u_i, v_i, 1)^T$ is its position in the 2D image plane expressed in homogeneous coordinates, they are related with the following expression:

$$sU_i=C[R\ t]X_i \quad \text{with} \qquad C=\begin{bmatrix}\alpha & 0 & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}$$

where s is a arbitrary scale factor, $[R\ t]$ are the extrinsic parameters. These are the rotation and translation which relates the camera coordinate system with the scene coordinate system. C contains the intrinsic parameters where $(u_0,\ v_0)$ corresponds to the coordinates of the principal point, α and β are the scale factors of the image in u and v axes. Calibration process consist of compute both internal and external camera parameters. To compute them intermediate ones are defined. In this case they correspond to the elements of the projection matrix. This matrix is called M and it is defined as $M=C[R\ t]$. This involves that $sU_i=MX_i$. To obtain the projection matrix elements starting from a set of points' coordinates X_i and U_i, they are arranged in a matrix in the following way:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1\cdot x_1 & -u_1\cdot y_1 & -u_1\cdot z_1 & -u_1\\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1\cdot x_1 & -v_1\cdot y_1 & -v_1\cdot z_1 & -v_1\\ .. & .. & .. & .. & .. & .. & .. & .. & .. & .. & .. & ..\\ x_n & y_n & z_n & 1 & 0 & 0 & 0 & 0 & -u_n\cdot x_n & -u_n\cdot y_n & -u_n\cdot z_n & -u_n\\ 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -v_n\cdot x_n & -v_n\cdot y_n & -v_n\cdot z_n & -v_n \end{bmatrix}\cdot\begin{bmatrix} m_{11}\\ m_{12}\\ ..\\ m_{33}\\ m_{34}\end{bmatrix}=0$$

it can be called:

$$A\cdot m=0$$

where m is the column vector with the unknown elements of the projection matrix and n the number of points used in the calibration process. The projection matrix elements are the elements of the eigen vector connected with the smallest eigen value of the matrix A^TA [8].

Since the noise in the points' coordinate measurements, the solution is always false. In order to obtain more information from the calibration process an interval in which the parameter is included, is computed. One way to do it is explained in the following section.

3 Mean and Variance of Projection Matrix Elements

Mean and the variance of projection matrix elements depend on the calibration algorithm and the noise level of the points' coordinates and also their situation in the world. It is supposed that they are situated in such a way that the resulting matrix A^TA is well conditioned. The aim is to compute a covariance matrix Λ_m of dimensions $(12x12)$ of the estimated projection matrix. In this section a method to compute them is presented assuming that the calibration process is carried out computing the eigen vectors of the matrix A^TA. Computation of the covariance matrix is based on the following theorem [3] [10]. Given a symmetric matrix B of dimensions nxn, exists an orthogonal matrix H which accomplished the following expression:

$$H^{-1}BH=diag\{\lambda_1,\lambda_2,.....\lambda_n\}$$

where $diag\{\lambda_1, \lambda_2, \ldots\ldots \lambda_n\}$ is a diagonal matrix with the eigen values of B starting from the small one.

$$\lambda_1 < \lambda_2 \leq \ldots\ldots \leq \lambda_n$$

Each column of the matrix H is called eigen vector and it is associated with an eigen value $\lambda_1, \lambda_2, \ldots\ldots \lambda_n$.

$$H = \begin{bmatrix} h_1 & h_2 & \ldots & h_n \end{bmatrix}$$

Elements of B corrupted with noise are called B_e and then $B_e=B+\Delta_B$. The perturbation of the eigen values using the perturbation of the matrix B is defined with

$$\Delta_{\lambda n} = h_n^T \Delta_B h_n$$

the perturbation of the associated eigen vector is

$$\Delta_{hn} = H \Delta H^{-1} \Delta_B h_n$$

where Δ, for the first eigen vector is

$$\Delta = diag\left\{0, (\lambda_1 - \lambda_2)^{-1}, \ldots\ldots, (\lambda_1 - \lambda_n)^{-1}\right\}$$

This theorem allows computing the eigen vector perturbation given the perturbation of the initial matrix Δ_B. If the projection matrix m is estimated computing the eigen values of A^TA, this theorem allows to compute the perturbation of the estimated elements of the projection matrix. Vector m corresponds to the eigen value λ_1. In order to compute this perturbation, last equation is used. The eigen vector which contains the elements of the matrix projection m is the first one since it is associated with the small eigen value. The perturbation of the estimated vector m is

$$\Delta_m = H \Delta H^{-1} \Delta_{A^T A} m$$

If this equation is rearranged in order to put the perturbation of the matrix A^TA in the right side, in a column vector the following expression arises:

$$\Delta_m = H \Delta H^{-1} D \Delta_{A^T A}$$

where D are the elements of the vector m rearranged since the perturbation of $A'A$ is now in a column vector $\Delta_{A^T A}$.

$$D = \begin{bmatrix} m_1 I_{12} & m_2 I_{12} & m_3 I_{12} & .... & m_{12} I_{12} \end{bmatrix} \quad (1)$$

I represents a unit $(12x12)$ matrix. From now to the end of the paper $\Delta_{A^T A}$ and Δ_A represent the perturbation of the matrix A^TA and A but arranged as a column vector. In order to express Δ_m as a function of the perturbation of the vector Δ_A, it is necessary to express $\Delta_{A^T A}$ as a linear function of Δ_A. If $\Delta_{A^T A} = f(\Delta_A) = K \Delta_A$ then

$$\Delta_m = H \Delta H^{-1} D K \Delta_A \quad (2)$$

If $L=H\Delta H^{-1}DK$ then $\Delta_m=L\Delta_A$ and therefore the variance of the computed elements of the projection matrix will be

$$\Lambda_m = L^T \Delta_A{}^T \Delta_A L = L^T \Lambda_A L$$

These are the result of applying the theorem to the calibration process and rearrange the matrices to express Λ_m as a function of Λ_A. In order to use this theorem, several terms should be defined before. These are matrix K, the value of Δ_A and the covariance matrix Λ_A. They are defined in the following sections.

3.1 Interval Computation Method

Interval computation method can be summarized as follows. Using (1), (2) the relation between Δ_m and Δ_A is defined as

$$\Delta_m = H\Delta H^{-1} D(F+G)\Delta_A = L\Delta_A$$

where

$$\Delta = diag\left\{0, (\lambda_1 - \lambda_2)^{-1}, ...(\lambda_1 - \lambda_n)^{-1}\right\}$$

$$F = \begin{bmatrix} F_1 & F_2 & F_3 & ..... & F_{2n} \end{bmatrix}$$

$$G = \begin{bmatrix} G_1 & G_2 & G_3 & ..... & G_{2n} \end{bmatrix}$$

$$F_i = \begin{bmatrix} a_{i1} I_{12} & a_{i2} I_{12} & a_{i3} I_{12} & ..... & a_{i12} I_{12} \end{bmatrix}^T$$

$$G_i = \begin{bmatrix} a^T O_{12x11} & O_{12x12} & O_{12x12} & .... & O_{12x12} \\ O_{12x12} & O_{12x1} a^T O_{12x10} & O_{12x12} & .... & O_{12x12} \\ O_{12x12} & O_{12x12} & O_{12x2} a^T O_{12x9} & .... & O_{12x12} \\ .... & .... & .... & .... & .... \\ O_{12x12} & O_{12x12} & O_{12x12} & .... & O_{12x11} a^T \end{bmatrix}$$

Therefore the variance of the elements of the projection matrix is defined as

$$\Lambda_m = L^T \Delta_A{}^T \Delta_A L = L^T \Lambda_A L$$

where Λ_A is computed with

$$\Lambda_A = \begin{bmatrix} \Lambda_{A1} & O_{24x24} & O_{24x24} & ... & O_{24x24} \\ O_{24x24} & \Lambda_{A2} & O_{24x24} & ... & O_{24x24} \\ O_{24x24} & O_{24x24} & \Lambda_{A3} & ... & O_{24x24} \\ ... & ... & ... & ... & ... \\ O_{24x24} & O_{24x24} & O_{24x24} & O_{24x24} & \Lambda_{An} \end{bmatrix}$$

Once the variance of the projection matrix elements is computed, it is possible to propagate this variance to the camera parameters. If a camera parameter p is related with the matrix projection elements with a function $p=g(m)$, the variance of the parameter p is

$$\Lambda_p = \nabla g(\overline{m}) \Lambda_m \nabla g(\overline{m})^T$$

where $\nabla g(\overline{m})$ represents the partial derivative of the function g for the computed matrix projection. Taking into account that the perturbation is modeled with a Gaussian distribution, the interval in which the parameter is contained can be computed using 95% of confidence degree. This is computed multiplying the standard deviation by 2.

4 Experimental Results

In order to test how well the variance estimation works several simulations have been carried out and also interval with real data from images has been calculated.

4.1 Simulations with Synthetic Data

A set of points are generated in two planes, and a camera is situated in the scene as is shown in figure 1. The camera is situated 1 m away form the Y axis, with an angle of incidence of the camera optical axe of 45 degrees with the X-Y plane. The features of the virtual camera are U_0=300 V_0=200 α = 1580 β=1580.

The camera is in the scene and a chessboard of 20x20 points is generated. The images of the points in the scene are using the projection matrix formed with the ideal camera parameters. Afterwards a noise level is added to the scene points coordinates and to the image points' coordinates. With the corrupted coordinates, the camera calibration is done. An estimated value and an interval in which it is included, is computed. This interval is obtained with the 95% of the variance. Since the real value is known, it can be compared with the estimated value and the computed interval. The most representative results have been obtained changing the noise level in the points' coordinates, and the number of points used in the camera calibration process. Figure 2 shows the results for the internal parameter. Remaining camera parameters have similar behavior. The left side figure shows how the estimated interval changes with the noise level in the coordinates increase. In this case 50 points are used. Since the noise level increase, the residual error is bigger and the estimated interval increase also. The central figure shows the effects of the number of points used as input data in the calibration process. The noise level is constant to 1 pixel in the image coordinates and 1 millimeter in the scene points´ coordinates. If the number of points increases the computed solution is less deviated from the real one and therefore the residual error of the minimization decrease. As a consequence the interval in which the camera parameter is contained decrease also. The right side figure compares the computed interval with 10 points and 1 pixel and 1 millimeter noise level, with several camera calibrations using a Montecarlo simulation. Estimations are represented with one dot. Border straight lines shows mean of the intervals computed with each calibration. This figure shows that the interval computed with the previous analysis is correct because border lines include 95% of the estimated values for the internal parameter U_0. Therefore, the value of the camera parameters will be always in this interval. This information will be very useful in order to mix information from several methods. The behavior of remaining parameters is very similar to this one.

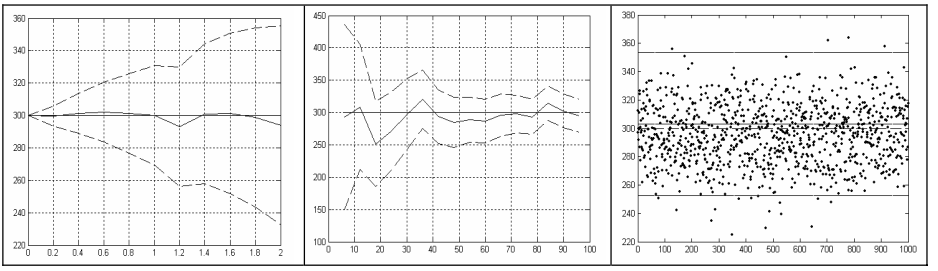
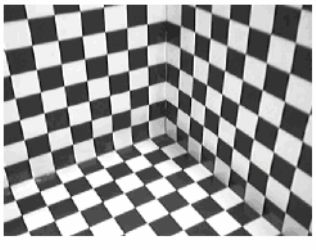


Fig. 1. Interval estimation of U_0. Straight line is the real value, continuous line is the estimated value and dotted line is the interval. Left – the noise level changes and the number of points is fixed. Center – the points number changes and the noise level is fixed. Right – the noise level and number of points is fixed and several camera calibrations have been performed. Each dot corresponds to one estimation. Border straight lines corresponds to the mean of interval computed with the presented method.

4.2 Experiments with Real Data

In this case images of a real calibration object has been taken. They are shown in figure 2. Camera is situated half a meter away from the origin of coordinates of the scene. About 150 points are used in each calibration process. Calibration is carried out and using the residual error of the minimization algorithm, the interval in which the camera parameters are contained is computed. Table attached to figure 2 shows the results of the calibration process with real data. The camera has been calibrated 100 times to test if the computed interval includes the true value. Right column shows the mean results together with its standard deviation and the left one the computed parameters and the computed uncertainty for one calibration process. Really, the computed interval includes the true value of the parameter.



	One Calibration	Mean Values
σ^2	1.124	1.027
U_0	329.248±12.429	317.259±1.648
V_0	232.268±15.521	245.157±0.574
α	1257.654±24.268	1279.681±1.876
β	1125.681±29.2658	1139±1.658
Tx	4.258±5.689	-0.158±0.268
Ty	79.267±10.598	75.684±4.589
Tz	68.268±8.547	73.895±3.258

Fig. 2. 3D pattern and estimated camera parameters and its intervals

5 Conclusions

A method to determine the variance of the projection matrix elements based on computing the eigen vector of the matrix $A'A$ has been presented. Using the residual error of the minimization process the standard deviation of the projection matrix

elements can be computed. From the point of view of 3D reconstruction process, interval in which the camera parameter is included is more useful than an accurate solution for a given set of data which is always false. Interval can be use to chose one or another 3D reconstruction method depending on the quality of the camera parameters and it can be computed with the presented method.

Acknowledgement

This work was financially supported by the Spanish government (CICYT project number DPI2004-08353-C03-02), European Community FEDER funds and European Community research funds (Project number 500095-2).

References

[1] J. R. Cooperstock, *Requirements for camera calibration: must accuracy come with high price?*. IEEE Work shop on applications of compueter vision. 2004
[2] O. Faugeras, *Three dimensional computer vision: A geometric viewpoint*. Cambridge, 1993.
[3] G.H.Golub, C.F.Van Loan, *Matrix computation, third edition.* The Jonh Hopkins University Press 1996
[4] R. I. Hartley, "In defence of the eight point algorithm". *IEEE Transactions on pattern analysis and machine intelligence.* 1997
[5] R.Hartley, A. Zisserman *Multiple view geometry in computer vision.* Cambridge 2000
[6] K. Kanatani, *Statistical Optimization for geometric computation,* Springer, June 1995.
[7] S.J. Maybank, O.D. Faugeras. "A theory of self-calibration of a moving camera". *The international Journal of computer vision*. 1992
[8] J. Salvi, X. Armangué, J. Battle *A comparative review of camera calibrating methods with accuracy evaluation*. Pattern Recognition 35, 2002
[9] J. Weng, P. Cohen, M. Herniou, "Camera calibration with distortion models and accuracy evaluation" *IEEE transactions on pattern analysis and machine intelligence.* 1992
[10] J. Weng, T.S. Huang, N. Ahuja, "Motion and structure from two perspective views: algorithms, error analysis and error estimation" *IEEE transactions on pattern analysis and machine intelligence.* 1989
[11] Z. Zhang, "A flexible new technique for camera calibration". *IEEE transactions on pattern analysis and machine intelligence* 2000
[12] Z. Zhang, "Calibration with one-dimensional objects". *Microsoft technical report*. 2002

A Comparison of 2-D Moment-Based Description Techniques

C. Di Ruberto and A. Morgera

Dipartimento di Matematica e Informatica,
Università di Cagliari, Cagliari, Italy

Abstract. Moment invariants are properties of connected regions in binary images that are invariant to translation, rotation and scale. They are useful because they define a simply calculated set of region properties that can be used for shape classification and part recognition. Orthogonal moment invariants allow for accurate reconstruction of the described shape. Generic Fourier Descriptors yield spectral features and have better retrieval performance due to multi-resolution analysis in both radial and circular directions of the shape. In this paper we first compare various moment-based shape description techniques then we propose a method that, after a previous image partition into classes by morphological features, associates the appropriate technique with each class, i.e. the technique that better recognizes the images of that class. The results clearly demonstrate the effectiveness of this new method regard to described techniques.

1 Introduction

Moment invariants were firstly introduced to the pattern recognition community in 1962 by Hu [1] , who employed the results of the theory of algebraic invariants and derived his seven famous invariants to rotation of 2-D objects. Since that time, numerous works have been devoted to various improvements and generalizations of Hu's invariants and also to its use in many application areas.

Dudani [2] and Belkasim [3] described their application to aircraft silhouette recognition, Wong and Hall [4], Goshtasby [5] and Flusser and Suk [6] employed moment invariants in template matching and registration of satellite images, Mukundan [7,8] applied them to estimate the position and the attitude of the object in 3-D space, Sluzek [9] proposed to use local moment invariants in industrial quality inspection and many authors used moment invariants for character recognition [3,10,11,12,13].

Maitra [14] and Hupkens [15] made them invariant also to contrast changes, Wang [16] proposed illumination invariants particularly suitable for texture classification. Li [17] and Wong [18] presented the systems of invariants up to the orders nine and five, respectively. Most recently, Flusser [19,20] has proposed a method how to derive independent sets of invariants of any orders.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 212–219, 2005.

There is also a group of papers [12,21,22] that use Zernike moments to construct rotation invariants.
In this paper we present a comparison among different description techniques based on moment invariants. Geometric and algebraic moments are described in section 2. We examine orthogonal moments in section 3. Section 4 discusses Generic Fourier Descriptor. In section 5 we test the described techniques for image indexing. In section 6 we propose a method to enhance the results obtained in previous experiments.

2 Geometric and Algebraic Moments

Moment invariants are useful in 2-D object recognition. Moment invariants are functions of moments that are invariant under certain transformations. Although moments are defined on a continuous image intensity function, a simple approximation is possible for a discrete binary image using summation operation. Let f be a binary digital image matrix with dimension M x N, and let

$$S = \{(x, y) | f(x, y) = 1\} \tag{1}$$

represent a 2-D shape. The *moment* of order *(p,q)* of shape S is given by

$$m_{pq}(S) = \sum\nolimits_{(x,y)\in S} x^p y^q. \tag{2}$$

The *central moment* of order *(p,q)* of shape S is given by

$$\mu_{pq}(S) = \sum\nolimits_{(x,y)\in S} (x - \bar{x})^p (y - \bar{y})^q \tag{3}$$

where $(\bar{x}, \bar{y})$ is the center of gravity. From the central moments, the normalized central moments have been defined. From the second- and third-order normalized central moments, a set of seven invariant moments, which is invariant to translation, scale change and rotation, has been derived by Hu [23]. Hu has also proved the invariance properties of the seven moments for the case of continuous function. Flusser and Suk [24] derived affine moment invariants wich are invariant under general 2-D affine tranformations. In [25] Taubin and Cooper defined algebraic moment invariants by introducing the concept of covariance matrix.

3 Orthogonal Moments

3.1 Zernike Moment Invariants

There are many different types of moments that have been applied to computer vision problems, but it has been demonstrated in [26] that the orthogonal Zernike moments offer a set of moments which are highly uncorrelated with little information redundancy. The orthogonal Zernike moments, first proposed by Teague

in [27], utilize the Zernike polynomial as basis function and are defined over the unit disc (in polar coordinates) by:

$$Z_{mn} = \frac{m+1}{\pi} \int_0^{2\pi} \int_0^1 V_{mn}^*(r,\theta) f(r,\theta) r dr d\theta \tag{4}$$

where m is the order of the moment (with $m \geq 0$) and n represents the repetition (where $|n| \leq m$, and $m + n$ is even). $V_{mn}(r,\theta)$ is the complex-valued Zernike polynomial with * indicating the complex conjugate. For a discrete square image (size Nx N), Z_{mn} can be calculated with:

$$Z_{mn} = \frac{m+1}{\pi} \frac{1}{(N-1)^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} V_{mn}^*(r,\theta) f(x,y) \tag{5}$$

given that $r = (x^2 + y^2)^{1/2}/\sqrt{2}N$ and $\theta = tan^{-1}(y/x)$, in order to map the image into the unit disc. Note that last equation is only orthogonal over the unit circle. The Zernike polynomial, $V_{mn}(r,\theta)$, is defined as:

$$V_{mn}(r,\theta) = R_{mn}(r) e^{-jn\theta} = R_{mn}(r)(\cos n\theta - j \sin n\theta) \tag{6}$$

where the radial polynomial, $R_m n(r)$, is:

$$R_{mn}(r) = \sum_{s=0}^{m-|n|/2} \frac{(-1)^s (m-s)! r^{m-2s}}{s! \left(\frac{m+|n|}{2} - s\right)! \left(\frac{m-|n|}{2} - s\right)!}. \tag{7}$$

This polynomial is such that over the unit disc, $|R_{mn}(r)| \leq 1$ and that $R_{mn}(1) = 1$, for any values of m and n. The definition of the radial polynomial also leads to $R_{mn}(r) = R_{m,-n}(r)$. The number of Zernike moments for any order, m, is given by $m + 1$, while the number of moments up to and including order m is $(m/2 + 1)(m + 1)$, (although because of the relationship between Z_{mn} and $Z_{m,-n}$ given above, only the moments with $n \geq 0$ need to be known).

3.2 Legendre Moments

The kernel of Legendre moments is products of Legendre polynomials defined along rectangular image coordinate axes inside a unit circle. The *(p,q)* order Legendre moments are defined as:

$$\lambda_{pq} = \frac{(2p+1)(2q+1)}{4} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} P_p(x) P_q(y) f(x,y) dx dy \tag{8}$$

where the function $P_p(x)$ denote Legendre polynomial of order p. The Legendre moment generalizes the geometric moments in the sense that the monomial $x^p y^q$

is replaced by the orthogonal polynomial $P_p(x)P_q(y)$ of the same order. The discrete version of the Legendre moments can be written as:

$$\lambda_{pq} = \frac{(2p+1)(2q+1)}{(M-1)(N-1)} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} p_p(x) p_q(y) f(x,y) \tag{9}$$

where *(p+q)* is the order, p,q=0, 1, 2, 3, ..., ∞. The Legendre polynomials, $P_p(x)$ are a complete orthogonal basis set on the interval [-1,1]:

$$\int_{-1}^{1} p_p(x) p_q(y) dx = \frac{2}{2p+1} \delta_{pq}. \tag{10}$$

The $n^{th} - order$ Legendre polynomial are defined by:

$$p_q(x) = \frac{1}{2^q} \sum_{p=0}^{\frac{q}{2}} (-1)^q \frac{(2q-2p)!}{p!(q-p)!(q-2p)!} x^{q-2p}. \tag{11}$$

4 2-D Generic Fourier Descriptors

Fourier transform has been widely used for image processing and analysis. The advantage of analyzing image in spectral domain over analyzing shape in spatial domain is that it is easy to overcome the noise problem which is common to digital images. At first a polar transformation of an input image *f(x,y)* is done, obtaining a polar image $f(\rho, \theta)$ by

$$r = \sqrt{(x-x_c)^2 + (y-y_c)^2}, \theta = tan^{-1}(y/x) \tag{12}$$

where (x_c, y_c) are the coordinates of centroid. Then a transformation of polar raster sampled image in Cartesian space is done. Then 2-D Fourier Transform of this polar raster sampled image $f(\rho, \theta)$ is calculated:

$$PF_2(\rho, \phi) = \sum_r \sum_i f(r, \theta_i) \exp\left[j2\pi \left(\frac{r}{R}\rho + \frac{2\pi i}{T}\phi \right) \right] \tag{13}$$

where $(0 \leq l < R)$ and $\theta_i = i(2\pi/T)(0 \leq i < T), 0 \leq \rho < R, 0 \leq \phi < T$. R and T are the resolution of radial frequency and angular frequency respectively. The normalized Fourier coefficients are the GFD.

5 Test of Retrieval Effectiveness

In order to test the retrieval performance of the described techniques we have constructed an image database made of 19 query shapes and 219 test shapes, composed by tools, biological and common shapes [28,29]. In figure 1 the experiment images are showed : they are numbered from 1 to 19 starting from top left corner. To recognize the test images we applied the described techniques according to the following 5 steps:

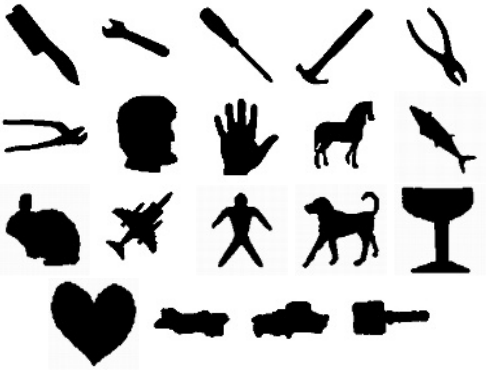


Fig. 1. Experiment shapes

1. We first create a feature vector for each query shape and each description technique. In particular, we construct such a vector by computing invariants for the image using geometric and algebraic moments based techniques. Applying orthogonal methods and GFD we obtain a coefficient matrix; in order to derive the feature vectors a linearization of this matrix for each image is done.
2. In the same way we create a feature vector for each test image.
3. By normalizing or not the feature vector, we compute the distance between the query shape feature vector and each test image feature vector; this distance has been computed both with Euclidean norm and with city-block norm. At the end of this step we have 4 measures.
4. Using a minimum distance criterion, we sort test images for each query shape.
5. If we indicate with M the number of shapes of a certain class, we can calculate a hit ratio as follows:
$$HR = \frac{\sum_{i=1}^{M} \alpha_i}{M} \tag{14}$$
where $\alpha_i = 1$ if the image belongs to that class, 0 otherwise.
6. As we use 4 measure distances, we obtain 4 hit ratios from which we calculate an average hit ratio for each query shape.

In table 1 you can see the indexing results. To consider higher order moments with orthogonal moments and GFD we have mediated among order mn and $2 \leq m,n \leq 7$ with $n \leq m$. Computing all mn orders you can obtain the maximum hit ratio for each method : Legendre 39.04%, Zernike 57.95%, GFD 60.13%. It's important to observe that we are not too far from the global medium hit ratio. As we can notice from results, Flusser's invariants and Legendre moments have a low capability of shape recognition if we compare them to other tested methods. This fact is due to inherent limitation; in fact Legendre moments are not invariants under image rotation while Flusser's invariants worsk only with affine transformation [24].

Table 1. 19 query shape experiment results

	HU	FLU	TAU	LEG	ZER	GFD
img1	0.3333	0.0000	0.1111	0.0972	0.4167	0.1389
img2	0.4500	0.0500	0.6000	0.1250	0.5000	0.5542
img3	0.5577	0.1923	0.3846	0.0961	0.3462	0.4808
img4	0.5500	0.2000	1.0000	0.4667	0.4000	0.4833
img5	0.0769	0.3077	0.3847	0.0994	0.3654	0.5705
img6	0.5000	0.2000	0.2000	0.0750	0.6500	0.3167
img7	0.2941	0.0882	0.1912	0.1814	0.3088	0.2500
img8	0.6500	0.2000	0.6500	0.3250	1.0000	0.9167
img9	0.6346	0.4231	0.3077	0.0449	0.2308	0.1410
img10	0.4000	0.4000	0.1000	0.3792	0.3000	0.6792
img11	0.4500	0.3500	0.5750	0.1167	0.5250	0.7958
img12	0.3250	0.1000	0.4750	0.3458	0.6750	0.5250
img13	0.3250	0.4500	0.2250	0.6250	0.3000	0.8542
img14	0.3269	0.3462	0.1539	0.2532	0.3462	0.2212
img15	0.7500	0.7273	0.5455	0.6212	0.8182	0.8977
img16	0.3863	0.2727	0.7273	0.5909	0.9546	0.7311
img17	0.5910	0.8182	0.9091	0.6137	0.8637	0.8788
img18	0.9091	0.8182	0.7955	0.7652	0.9091	0.9621
img19	0.4773	0.1818	0.7500	0.3977	0.9318	0.9205
global	8.9872	6.1256	9.0855	6.2192	10.8413	11.3175
perc	47.30	32.24	47.82	32.73	57.06	59.57

6 Methods Enhancement

In order to improve the performance of the described methods, we partition images into classes so that we can associate a proper shape recognition method with each identified class. To identify the class partitioning we extract effective features of images. To do this, we consider morphological image features and other information derived from the image skeleton. We use then the following features :

- Normalized area
- Elongatedness
- Circularity
- Skeleton branch number

After feature normalization in [0:1] interval, we divide images into 6 classes with a K-means clustering algorithm. For each image we identify the method that better recognizes this shape by calculating the best medium hit ratio. Combining this choice with clustering, we decide to select a method for that class based on majority criterion. In tie case we do a minimum variance choice. For example the k-means classification assign the same class to images n.5 and n.6. The best results we obtained for image n.5 is with GFD, while for image n.6 is obtained with Zernike moments. In this case we choose Zernike moments because we obtained a lower variance. In table 2 we show numerical values of the experiment.

Table 2. Resuming table of normalized image features, classification classes, chosen description technique

img N.	Normalized Area	Elongatedness	Circularity	# Skel. Branch	Class_#	Method
1	0.2298	0.4937	0.1150	0.1667	1	TAU
2	0.0000	0.5511	0.2502	0.0833	1	TAU
3	0.0174	1.0000	0.2672	0.0833	1	TAU
4	0.0623	0.6720	0.3553	0.0000	1	TAU
5	0.0748	0.1722	0.7309	0.1667	2	ZER
6	0.0667	0.1662	1.0000	0.1667	2	ZER
7	0.7484	0.0197	0.0249	0.6667	3	GFD
8	0.4464	0.0308	0.4183	0.5833	4	GFD
9	0.2548	0.0819	0.6590	1.0000	5	ZER
10	0.0695	0.4496	0.2908	0.5000	4	GFD
11	0.6254	0.0700	0.0586	0.8333	3	GFD
12	0.2130	0.0409	0.4360	0.8333	5	ZER
13	0.1986	0.0491	0.5046	0.2500	4	GFD
14	0.4471	0.0000	0.7777	0.6667	5	ZER
15	0.5845	0.0571	0.3412	0.1667	4	GFD
16	1.0000	0.0246	0.0000	0.5000	6	ZER
17	0.9060	0.2681	0.1051	0.1667	6	ZER
18	0.8145	0.1727	0.0969	0.6667	3	GFD
19	0.7535	0.1776	0.1795	0.2500	6	ZER

From this classification we obtain a global hit ratio of 12.4688, in percentage 65.63% versus the best result we obtained with previous methods, 59.57% of GFD. It is important to notice that the best result we can obtain, that is to choose for each image the best method that recognizes it, is about 75.90%.

7 Conclusion

In this paper we introduced invariant features that may be used for shape based image retrieval. We considered Generic Fourier descriptors, Hu's moment invariants, Taubin's moment invariants Flusser's moment invariants, Zernike moments and Legendre moments. Classification based on morphological image features, combined with moment invariants described, showed effective image retrieval.

References

1. M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, 8:179–187, 1962.
2. S. A. Dudani, K. J. Breeding, and R. B. McGhe. Aircraft identification by moment invariants. *IEEE Trans. Computers*, 26:39–45, 1977.
3. S. O. Belkasim, M. Shridhar, and M. Ahmadi. Pattern recognition with moment invariants : a comparative study and new results. *Pattern Recognition*, 24:1117–1138, 1991.
4. R. Y. Wong and E. L. Hall. Scene matching with invariant moments. *Computer Graphics and Image Processing*, 8:16–24, 1978.

5. A. Goshtasby. Template matching in rotated images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7:338–344, 1985.
6. J. Flusser and T. Suk. A moment-based approach to registration of images with affinne geometric distortion. *IEEE Trans. Geoscience and Remote Sensing*, 32:382–387, 1994.
7. R. Mukundan and K. R. Ramakrishnan. An iterative solution for object pose parameters using image moments. *Pattern Recognition Letters*, 17:1279–1284, 1996.
8. R. Mukundan and N. K. Malik. Attitude estimation using moment invariants. *Pattern Recognition Letters*, 14:199–205, 1993.
9. A. Sluzek. Identification and inspection of 2-d objects using new moment-based shape descriptors. *Pattern Recognition Letters*, 16:687–697, 1995.
10. F. El-Khaly and M. A. Sid-Ahmed. Machine recognition of optically captured machine printed arabic text. *Pattern Recognition*, 23:1207–1214, 1990.
11. K. Tsirikolias and vol. 26 pp. 877-882 1993. B. G. Mertzios, ". Statistical pattern recognition using efficient two-dimensional moments with applications to character recognition. *Pattern Recognition*, 26:877–882, 1993.
12. A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12:189–197, 1990.
13. J. Flusser and T. Suk. Affine moment invariants: A new tool for character recognition. *Pattern Recognition Letters,*, 15:433–436, 1994.
14. S. Maitra. Moment invariants. *Proc. of the IEEE*, 67:697–699, 1979.
15. T. M. Hupkens and J. de Clippeleir. Noise and intensity invariant moments. *Pattern Recognition*, 16:371–376, 1995.
16. L. Wang and G. Healey. Using zernike moments for the illumination and geometry invariant classiffication of multispectral texture. *IEEE Trans. Image Processing*, 7:196–203, 1998.
17. Y. Li. Reforming the theory of invariant moments for pattern recognition. *Pattern Recognition*, 25:723–730, 1992.
18. W. H. Wong, W. C. Siu, and K. M. Lam. Generation of moment invariants and their uses for character recognition. *Pattern Recognition Letters*, 16:115–123, 1995.
19. J. Flusser. On the independence of rotation moment invariants. *Pattern Recognition*, 33:1405–1410, 2000.
20. J. Flusser. On the inverse problem of rotation moment invariants. *Pattern Recognition*, 35:3015–3017, 2002.
21. M. R. Teague. Image analysis via the general theory of moments. *J. Optical Soc. of America*, 70:920–930, 1980.
22. A. Wallin and O. Kubler. Complete sets of complex zernike moment invariants and the role of the pseudoinvariants. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17:1106–1110, 1995.
23. M. K. Hu. Pattern recognition by moment invariants. *Proc.IRE*, 49, 1961.
24. J. F. T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26:167–174, 1993.
25. G. Taubin and D.B. Cooper. *Geometric invariance in computer vision*, chapter Object recognition based on moment (or algebraic) invariants, pages 375–397. MIT Press, 1992.
26. C-H. Teh and R.T. Chin. On image analysis by the method of moments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.
27. M.R. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70(8):920–930, 1979.
28. http://www.cs.rutgers.edu/pub/sven/rutgers-tools/.
29. http://www.lems.brown.edu/vision/software/.

A Compact System for Real-Time Detection of Line Segments

Nozomu Nagata and Tsutomu Maruyama

Systems and Information Engineering, University of Tsukuba,
1-1-1 Ten-ou-dai Tsukuba Ibaraki 305-8573 Japan
nagata@darwin.esys.tsukuba.ac.jp

Abstract. In this paper, we describe a compact circuit for real-time detection of line segments using the Line Hough Transform (LHT). The LHT is a technique to find out lines in an image. The LHT is robust to noises, but requires long computation time. The circuit calculates (1) r and θ of lines (r is the distance from the origin to a line and θ is the angle of the line) by the LHT units in parallel, and (2) start and end points of the lines by the other units which are completely pipelined with the LHT units. With this parallel and pipeline processing, the circuit can detects line segments by $\pi/512$ angle steps in a standard size image (640×480) in real-time. This circuit was implemented on an off-the-shelf PCI board with one Field Programmable Gate Array (FPGA) chip. The size of the circuit is 45% of the chip, which makes it possible to implement other circuits for higher level processing of object recognition on the same chip, or the performance can be improved twice by using the rest of hardware resources.

1 Introduction

Detection of line segments is a very important step in object recognition. Line segments with other informations such as distances to the planes surrounded by them will help to construct 3-D models of real-world. The Line Hough Transform (LHT) is a technique to find out r (distance from the origin to a line) and θ (the angle of the line) of lines in an image. The LHT is very robust to noises, but requires long computation time for calculating (r, θ) of all candidates. The calculation, however, is very regular and simple, and can be accelerated by parallel processing by hardware. In our circuit, r and θ of lines are detected by LHT units first, and then, start and end points of the lines are obtained by the other units which are completely pipelined with the LHT units. In these units, start and end points of lines are obtained by considering distances between edge points on the lines. This parallel and pipeline processing make it possible to find out start and end points of lines in standard size images (640×480) in real-time.

2 Previous Works

Many approaches by software to reduce the computational complexity have been proposed, and efficient algorithms to extract line segments have also been re-

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 220–228, 2005.

searched, but it is still difficult to detect line segments in real-time by one micro-processor.

In order to accelerate the LHT by hardware, many systems have been developed to date [1-11]. They can be categorized into three groups; parallel systems with many processing elements, dedicated hardware systems with ASICs, and reconfigurable systems with Field Programmable Gate Arrays (FPGAs). Of these three approaches, systems on reconfigurable devices are most promising at the present time because the performance of one reconfigurable device such as an FPGA is enough to realize real-time processing of the LHT, and the systems with reconfigurable devices can change their functions according to their circumstances.

In our system, (1) line segments (start and end points of lines) are detected (not only the lines by the LHT), and (2) more than 16348 line segments can be found in a standard size image in real-time, though it is far beyond the practical requirement, and not useful.

3 The Line Hough Transform

In this section, we overview the Line Hough Transform (LHT). In the LHT, a line is given by r (distance to the line from the origin) and θ (the angle of the line). In Figure 1(a), a line that goes through a point (x_i, y_i) is shown by the equation below.

$$r_k = x_i \times cos\theta_k + y_i \times sin\theta_k$$

In the LHT, when an edge point is given, all lines that go through the edge point are considered as candidates of the true line. In order to calculate (r, θ) of all the lines, the range of θ $(0 - \pi)$ is divided by N, and $(r_n, n/N \times \pi)$ $_{n=(0,N-1)}$ are calculated. With larger N, we can obtain more precise r and θ, but it requires more amount of memory and computation time. In Figure 1(b), three curves $(r = x_i cos\theta + y_i sin\theta)$ are plotted on (r, θ) plane, and each point on these curves shows a line that goes through (x_i, y_i). Among all points on these curves, the crossing points of the curves $((r_k, \theta_k))$ gives the true line.

In order to find the crossing points, for each edge point, N pairs of (r_i, θ_i) $(\theta = \pi \times i/N,\ i = 0, N-1)$ are calculated, and the value in (r, θ) memory are incremented using (r_i, θ_i) as addresses as shown in Figure 1(c). Then, peak values in (r, θ) memory are chosen as true lines in (x, y) plane.

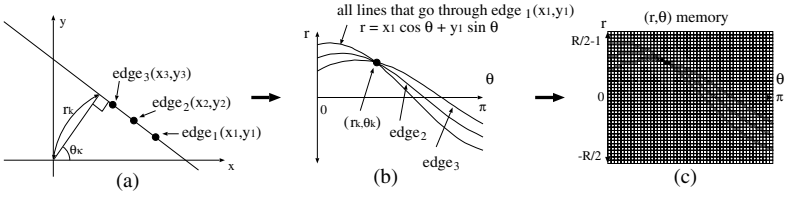


Fig. 1. The Outline of the LHT

4 Detection of Line Segments

In this section, we introduce the hardware algorithm to obtain line segments. In our approach, the following procedures are repeated twice to simplify memory access pattern and achieve higher performance.

1. Detect edge points in an given image.
2. Apply the Line Hough Transform (LHT) to the edge points.
3. Find start and end points of lines obtained by the LHT.

In the first path (*vertical search*), line segments in $\theta = [-\pi/4, \pi/4)$ are detected, while those in $\theta = [\pi/4, \pi \times 3/4)$ are detected in the second path (*horizontal search*). In our approach, $[-\pi/4, \pi \times 3/4)$ is used instead of $[0, \pi)$ as the range of θ. In *the vertical search*, data in the given image are read out along x axis (Figure 2(a)), while they are read out along y axis (Figure 2(b)) in *the horizontal search* in order to simplify non-maximum suppression in the edge detection, and discontinuity check in finding start and end points of line segments.

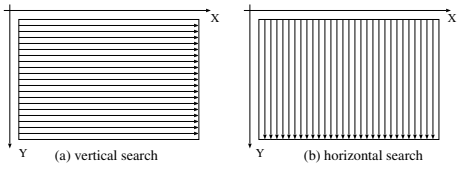


Fig. 2. The Scan Direction of Data in the Image

4.1 Edge Detection

First, the magnitude $M(x, y)$ of each pixel is calculated by Sobel filter.

$$M(x, y) = |G_h(x, y)| + |G_v(x, y)|$$

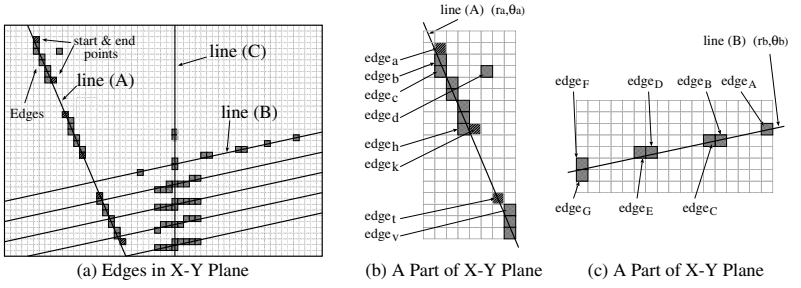
Then, non-maximum suppression is applied. In *the vertical search*, data in the image are read out along x axis as shown in Figure 2(a), and $M(x, y)$ is suppressed if $M(x, y)$ is smaller than $M(x - 1, y)$ or $M(x + 1, y)$. In *the horizontal search*, data are read out along y axis, and $M(x, y)$ is suppressed if $M(x, y)$ is smaller than $M(x, y - 1)$ or $M(x, y + 1)$. In our approach, orientation of an edge point is not calculated using G_v/G_h, but the search itself is divided by the orientation.

4.2 The Line Hough Transform

Suppose that the size of the image is $X \times Y$, and the range of θ is divided by N. Then, the computation order of the LHT becomes $X \times Y \times p \times N$, where p is the rate that a pixel in the image is an edge point. In this computation, the maximum parallelism can be N. The performance by parallel processing with N units and N memory banks is much faster than the video-rate in general. Therefore, parallel processing by N/m units is the best approach to achieve real-time processing with a smaller circuit.

4.3 Detection of Start and End Points of Lines

In Figure 3(a), line (A) goes on three line segments. In this case, only line (A) is detected by the LHT, and the three line segments can not be distinguished. Furthermore, line (B) and (C) that go through non-continuous edge points are also detected as lines because they make peaks in (r, θ) memory.



(a) Edges in X-Y Plane (b) A Part of X-Y Plane (c) A Part of X-Y Plane

Fig. 3. Lines Obtained by the LHT

Start and end points of lines can be found by checking pixels near the lines obtained by the LHT whether they are edge points or not. This approach is, however, not suitable for hardware systems because it requires non-regular memory accesses. We need a method to find start and end points of lines (1) which can be completely pipelined with the computation of the LHT, and (2) which does not require non-regular memory accesses.

A Procedure to Find out Start and End Points of Lines. In our method, four variables (*Start Point* (x, y), *Last Point* (x, y), *Counter* and *IsLine*) are given to each peak in (r, θ) memory. Using these variables, start and end points of lines are found as follows.

1. All *Counters* are initialized to zero.
2. For each edge point in the image, (r_i, θ_i) are calculated again.
3. Values in (r, θ) memory are read out using (r_i, θ_i) as addresses, and if (r_i, θ_i) is a peak,
   (a) If *Counter* is zero,
      i. the address of the edge point is set to *Start* and *Last Point*, and
      ii. *Counter* is incremented.
   (b) else (*Counter* is not zero)
      i. The distance between *Last Point* and the edge point is calculated.
      ii. *Counter* is incremented/decremented according to the distance, and
         A. If *Counter* becomes larger than a threshold, 1 is set to *IsLine*, and the threshold is set to *Counter*.
         B. If *Counter* becomes negative,
            – if *IsLine* is one, *Start* and *Last Point* are output as start and end points of a line,

– else, nothing is output.
Counter and *IsLine* are reset, and the address of the edge point is set to *Start Point*.

iii. The address of the edge point is set to *Last Point*.

4. After processing all edge points, (r, θ) memory is scanned, and if peaks whose *IsLine* are one are found, their *Start* and *Last Points* are output as start and end points of lines.

With this method, the computation to find start and end points can be completely pipelined with the computation of the LHT, because both computations use same input (same edge points) and finish in exactly same clock cycles.

In Figure 3(b), one of (r_i, θ_i) for $edge_a$ hits (r_a, θ_a), which is a peak in (r, θ) memory. Then, the address of $edge_a$ is set to its *Start* and *Last Point*, and *Counter* is incremented, because *Counter* was initialized to zero. In the computation of $edge_b$, *Counter* is incremented, because *Counter* is not zero, and the distance between *Last Point* $(edge_a)$ and $edge_b$ is zero. The address of $edge_b$ is set to *Last Point*. As for $edge_d$, no (r_i, θ_i) hits peaks in (r, θ) memory, and no variables on peaks are changed. By repeating the computation, *Counter* in (r_a, θ_a) becomes larger than the threshold, and its *IsLine* becomes one. When $edge_t$ is processed, the distance between *Last Point* $(edge_k)$ and $edge_t$ is large, and *Counter* becomes negative. Then, *Start Point* $(edge_a)$ and *Last Point*$(edge_k)$ are output as start and end points of a line. *Counter* and *IsLine* are reset, and $edge_t$ is set to *Start* and *Last Point*. In Figure 3(c), *Counter* in (r_b, θ_b) does not exceed the threshold, and nothing is output.

A Technique to Find Line Segments Under Noises. As shown in Figure 4(a), the actual edge points do not form an ideal line. Figure 4(b) shows a part of (r, θ) memory which are incremented by the edge points in Figure 4(a). In Figure 4(b), (r_k, θ_k) is chosen as a peak, but it is not sharp. In this case, $(r_i, \theta_i)_{\ i=0,N-1}$ calculated from the edge points do not always hit (r_k, θ_k) as shown in Figure 4(c). In Figure 4(c), only edge points in dark gray hit (r_k, θ_k), and other edge points hit $(r_k \pm l, \theta_k)_{\ l=1,\ldots}$. Therefore, we need a technique to consider $(r_k \pm l, \theta_k)_{\ l=0,\ldots}$ as one peak.

Figure 5 shows a technique to recognize $(r_k \pm l, \theta_k)_{\ l=0,\ldots}$ as one peak. In Figure 5, another memory (*R-Translation Table*) which has the same number of entries with (r, θ) memory is used, and when peaks in (r, θ) memory are found, address of *Start & End Points Table* is stored at $(r_k \pm l, \theta_k)$ in the *R-Translation*

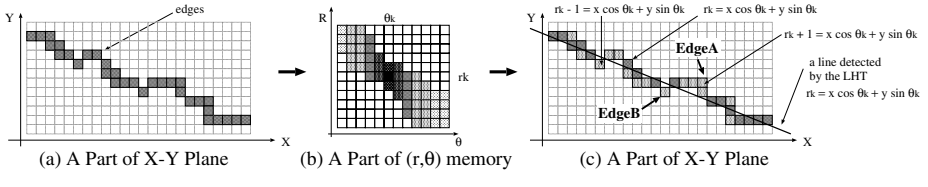


Fig. 4. Lines Obtained By Filtering

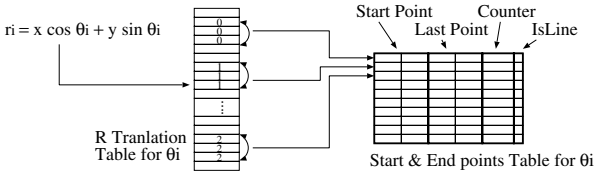


Fig. 5. R Translation Table

Table ($l = 0, 1$ in Figure 5). The address of *Start & End Points Table* starts from one (address zero means that it is not a peak), and incremented when a new peak is found in (r, θ) memory. When r_i for θ_i is calculated from an edge point, *R-Translation Table* is looked up, and if the value on (r_i, θ_i) is not zero, *Start & End Points Table* is accessed using the value as an address.

Distance Between Edge Points. In the technique described above, the scan direction of edge points is very important. In Figure 4(c), suppose that edge points are read out along X axis. Then, *EdgeB* hits (r_k, θ_k) next to *EdgeA*. These two edge points are not continuous, but we need to consider these points are continuous. This judgment becomes more difficult as θ of the line becomes closer to $\pi/2$, because the line becomes more parallel to X axis. Therefore, we need *the vertical and horizontal search* with different scan direction of edge points. In *the vertical search*, only y is used to decide the continuity of edge points, while only x is used in *the horizontal search*.

4.4 Parallel and Pipeline Processing

The sequence of the line segment detection described in the previous subsections can be summarized as follows.

1. Find edge points.
2. Calculate r_i, and increment values in (r, θ) memory.
3. Find peaks in (r, θ) memory.
4. Find start and end points of lines (start and end points of some line segments are output).
5. Output start and end points of lines scanning *Start & End Points Tables*.

In our implementation, N is 512. This means that θ is divided by 512, and 512 r are calculated for each edge point. By using smaller N, we can reduce computation time, but we can not get (r, θ) which matches well with long lines on the standard size image. The size of the (r, θ) memory becomes 512×800 ($N \times length\ of\ the\ diagonal$).

Figure 6 shows the parallel and pipeline processing realized in our circuit to achieve real-time visualization. In Figure 6,

1. addresses of edge points are generated by *Edge Detection Unit* and stored in an external memory.
2. the addresses of edge points are read out $(16 + 1) \times 2$ times.

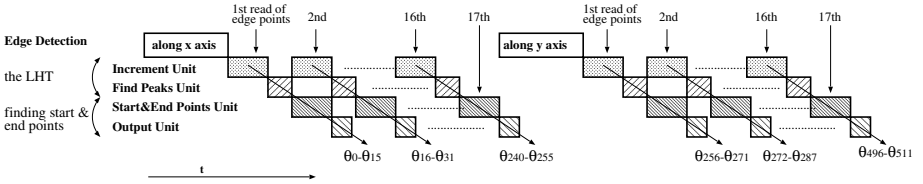


Fig. 6. Pipeline Processing

3. 16 θ_i are processed in each pipeline step with 8 units which run in parallel. Each unit processes two θ_i sequentially.
4. $(4+8+4)$ units run in parallel in the LHT in order to find peaks at least in the range of $(\theta_{i-4}, \theta_{i+4})$.
5. *Increment Unit* and *Start & End Point Unit* can not be active with *Find Peak Unit* and *Output Unit* at the same time, because they share same memory banks.

5 Performance

This circuit was implemented on an off-the-shelf PCI board (ADM-XRC-II by Alpha Data [13]) with one FPGA chip (Xilinx XC2V6000 [14]). The circuit runs at 66 MHz, and 45% of logic blocks, 33% of multipliers, and 29% of internal memory banks are used.

Table 1 shows the performance when the rate that a pixel in an image is an edge points is 5%, 7% and 9%. As shown in Table 1, the performance depends on the rate (the number of edge points in an image). The rate can be controlled by a threshold which is used to decide whether a pixel is an edge point in *Edge Detection Unit*. Lower threshold generates more edge points, but more noises are included. In this sense, the circuit can process more than 30 frames in one second under proper range of the threshold.

Figure 7 and 8 show the original image, and the edge points detected in *the vertical search*. Our algorithm is still sensitive to some parameters such as a threshold in the edge detection, the minimum peak height in the LHT, minimum length of line segments, and l (the parameter to recognize several (r, θ) as one peak described in 4.3). Figure 9 and 10 show the line segments detected by our circuit, when $l = [0, 1]$ and $[-1, 1]$ respectively. In Figure 9, all detected line segments fit well with the original image, but we can not detect long line segments because of the distortion of the camera and noises. In Figure 10, we can

Table 1. Performance of the Circuit

rate of edge points	clock cycles per image	frame per second
5%	1713290	38.9
7%	2131072	31.3
9%	2548864	26.2

Fig. 7. Original Image

Fig. 8. Edge Points (Vertical Search)

Fig. 9. Line Segment ($l = [0, 1]$)

Fig. 10. Line Segments ($l = [-1, 1]$)

detect long line segments, but some short lines do not fit well with the original image because of noises. We may need to control these parameters according to the length of line segments, though it requires more computation time.

6 Conclusions

In this paper, we described a compact circuit for real-time detection of line segments using the Line Hough Transform. This circuit was implemented on an off-the-shelf FPGA board with one FPGA chip. The circuit can find start and end points of line segments in an image (640×480 pixels) in real-time (more than 30 images per second). 45% of the FPGA chip is used for the circuit. With the rest of hardware resources, we can implement other circuits for higher level processing of object recognition, or improved the performance twice.

We are now improving the circuit to make it more robust to noises, and more insensitive to the parameters.

References

1. K. Hanahara, T. Maruyama and T. Uchiyama, "A real time processor for the Hough transform", IEEE Trans. Pattern. Anal. Mach. Intell. 10 (1987), pp. 121-125.
2. F.M. Rhodes, J.J. Disturi, G.H. Chapman, B.E. Emerson, A.M. Soares and J.I. Raffel, "A monolithic Hough transform processor based on restructurable VLSI", IEEE Trans. Pattern. Anal. Mach. Intell. 10 (1988), pp. 106-110.
3. M.F.X.D Van Swaaij, F. V. M. Catthoor and H. J. De Man, "Deriving ASIC architecture for the Hough Transform", Parallel Computing 16, 1990, pp.113-121
4. M.Atiquzzamau, "Pipelined implementation of the multi-resolution Hough Transform in a pyramid multiprocessors", Pattern Recognition Letters, 1994, pp.841-851.
5. D. Ben-Tzvi, A. Naqui, M. Sandler, "Synchronous multiprocessor implementation of the Hough Transform", Computer Vision Graphics Image Process 1990, pp.437-446.
6. A-N. Choudhary and R.Ponnussary, "Implementation and evaluation of Hough transform algorithm on shared-memory multiprocessors", J. Parallel Distributed Comput. 12, 1991, pp.178-188.
7. A. L. Abbott, P. M. Athanas, L. Chen, and R. L. Elliott, "Finding Lines and Building Pyramids with Splash 2", FCCM 1994.
8. Chung, K.L., Lin, H.Y., "Hough Transform On Reconfigurable Meshes", Computer Vision and Image Processing, No. 2, March 1995, pp. 278-284.
9. M. Nakanishi and T. Ogura, "Real-time line extraction using a highly parallel Hough transform board", Proceedings of International Conference on Image Processing, 1997, pp. 582-585.
10. Pan, Y., Li, K., Hamdi, M., "An Improved Constant-Time Algorithm for Computing the Radon and Hough Transforms on a Reconfigurable Mesh", IEEE Trans. Systems, Man and Cybernetics-A(29), No. 4, July 1999, pp. 417.
11. Tagzout, S., Achour, K., Djekoune, O., "Hough transform algorithm for FPGA implementation", Signal Processing 81, No. 6, June 2001, pp. 1295-1301.
12. N. Nagata and T. Maruyama, "Real-time Detection of Line Segments Using The Line Hough Transform", IEEE International Conference on Filed-Programmable Technology, 2004, pp. 89 - 96.
13. http://www.alpha-data.com
14. http://www.xilinx.com

Discrete 3D Tools Applied to 2D Grey-Level Images

Gabriella Sanniti di Baja[1], Ingela Nyström[2], and Gunilla Borgefors[3]

[1] Institute of Cybernetics "E.Caianiello", CNR, Pozzuoli, Italy
gsdb@imagm.cib.na.cnr.it
[2] Centre for Image Analysis, UU, Uppsala, Sweden
ingela@cb.uu.se
[3] Centre for Image Analysis, SLU, Uppsala, Sweden
gunilla@cb.uu.se

Abstract. 2D grey-level images are interpreted as 3D binary images, where the grey-level plays the role of the third coordinate. In this way, algorithms devised for 3D binary images can be used to analyse 2D grey-level images. Here, we present three such algorithms. The first algorithm smoothes a 2D grey-level image by flattening its geometrical and grey-level peaks while simultaneously filling in geometrical and grey-level valleys, regarded as non significant in the problem domain. The second algorithm computes an approximation of the convex hull of a 2D grey-level object, by building a covering polyhedron closely fitting the corresponding object in a 3D binary image. The result obtained is convex both from the geometrical and grey-level points of view. The third algorithm skeletonizes a 2D grey-level object by skeletonizing the top surface of the object in the corresponding 3D binary image.

1 Introduction

Algorithms for 3D voxel images are generally regarded as computationally more expensive than algorithms for 2D pixel images. This is certainly at least partially true both due to the large number of elements generally constituting a voxel image, and due to that a number of geometrical and topological problems arise when working with 3D objects that are not present in 2D space. However, there are cases when 3D algorithms can be conveniently used, even though the images to be analysed belong to the 2D space. This is the case, in our opinion, when working with 2D grey-level images. A 2D grey-level image can be interpreted as a terrain elevation map, where the grey-level of a pixel is interpreted as its height with respect to the grey-level of the background. By using this interpretation, we can pass from the input 2D grey-level image to a 3D binary image. In this way, algorithms developed for treating binary images can be used. These are generally computationally less expensive than algorithms taking into account also grey-level information. Moreover, some of the processing to be performed will be concerned mainly with the voxels belonging to the top surface of the so obtained solid object, which limits the number of elements to be processed notwithstanding the fact that generally a rather large number of voxels constitutes the 3D binary image corresponding to the 2D grey-level image.

In this paper, we convert a 2D grey-level image to a 3D binary image and present some 3D binary image analysis tools intended for analysing the 2D grey-level image. We assume that the grey-level image is already segmented, so that the regions corresponding to the grey-level objects can be distinguished from a uniform background.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 229 – 236, 2005.

We modify the algorithm suggested in [1] to smooth 3D objects in binary images, to actually smooth a 2D grey-level object. Geometrical and grey-level peaks of the object, regarded as non significant in the problem domain, are flattened and non significant geometrical and grey-level valleys are simultaneously filled in. The smoothing algorithm is based on the standard concept of erosion/dilation [2], but is here accomplished by using the distance transform of the object with respect to the background and the distance transform of the background with respect to the object. This smoothing algorithm is a useful tool per se and, in particular, it can be used before applying any of the remaining tools we introduce in this paper, namely convex hull approximation and skeletonization, to improve their performance.

We adapt the algorithm introduced in [3] to compute an approximation of the convex hull of 3D objects in binary images, to the case of 2D grey-level objects. A first attempt in this direction has recently been done [4]. The approximation of the convex hull is a covering polyhedron, closely fitting the object, obtained by iteratively filling local concavities in the 3D binary image. The obtained result is convex both from the geometrical point of view and as concerns grey-levels. The grey-level convex deficiency can be used to eliminate uneven illumination in grey-level images.

Using the algorithm suggested in [5] to compute the curve skeleton of surface-like objects in binary images and its modification introduced in [6], we present an algorithm to skeletonize 2D grey-level objects by actually skeletonizing the top surface of the 3D objects in the corresponding binary images. Voxels in the top surface of a 3D object are examined in the order of their distances from the initial border of the surface, which limits the computation time. Moreover, skeleton branches will be placed along significant (geometrical and grey-level) ridges of the 2D grey-level object, since the skeletonization of the 3D surfaces takes into account geometric information.

Our algorithms are based on the use of local operations and, except for skeletonization which requires a number of scans dependent on the size of the top surface of the object in the converted 3D image, all computations are done in a fixed and small number of scans. Therefore, they have reasonable computational complexity and short processing times on standard computers. We have tested our algorithms on a variety of images and the obtained results have been satisfactory. A comparison with other algorithms for 2D grey-level smoothing and skeletonization remains to be done. The method for a grey-level convex hull is unique.

2 Conversion Between 2D Grey-Level and 3D Binary Images

We start with an $M \times N$ 2D grey-level image G. For simplicity, we assume that only one object is present in G and that the background is uniform. The pixels in G belong to $[0,K]$, typically, $K = 255$. Object pixels range from 1 to K, while background pixels are 0. To build an $M \times N \times (K + 1)$ binary image I, for each pixel at position (x,y) in G with grey-level g, all voxels at positions (x,y,z) in I with $1 \leq z \leq g$ are set to 1 (object), while all voxels at positions (x,y,z) with $g + 1 \leq z \leq K$ and all voxels with z=0 are set to 0 (background). Note that since grey-level is not equivalent to height, the correspondence between grey-level g and z value is not given. It depends on the specific application. Linear scaling may be necessary. Each pixel (voxel) v in G (I) has two (three) types of neighbours called edge- or vertex- (or face-) neighbours, depending on whether they share an edge or a vertex (or a face) with v.

The obtained 3D image undergoes image analysis. Whenever the process is completed on the 3D binary image, the so resulting image can be converted to a 2D grey-level image by projection. This process is straightforward.

3 Grey-Level Image Smoothing

Smoothing can be done by erosion/dilation operations [2], or by distance transform based methods [1]. We favour the latter approach in particular as it allows us to smooth an object by simultaneously removing its protrusions, cavities and concavities of negligible size. To achieve this, the distance transform of the whole 3D image I is computed and, to save computation time, the distance transform of the object and the distance transform of the background are computed simultaneously.

Let w_f, w_e, and w_v denote the local distances from any voxel to its face-, edge-, and vertex-neighbours, respectively. The distance transform is computed by propagating local distance information during a forward and a backward scan of the image. Each inspected voxel v is set to the minimum of the value of the voxel itself and the values of its already visited neighbours increased by the local distances (w_f, w_e, or w_v) to the voxel v. Let *min* denote such a minimum. To distinguish between object and background, we ascribe positive sign to distance values computed for object voxels, and negative sign to distance values computed for background voxels. Initially, background voxels are set to -infinity and object voxels to +infinity. During the forward scan, every voxel v in the object (background) is set to:

w_f ($-w_f$), if v has a negative (positive) face-neighbour;
w_e ($-w_e$) if v has a negative (positive) edge-neighbour, but all its face-neighbours are positive (negative);
w_v ($-w_v$) if v has a negative (positive) vertex-neighbour, but all its face- and edge-neighbours are positive (negative);
min (*-min*), otherwise.

During the backward scan, only voxels with values different from $\pm w_f$, $\pm w_e$, and $\pm w_v$ are examined and are assigned value $\pm min$. Obviously, absolute values are taken into account to correctly compute the *min* for the background voxels.

The obtained distance transform of the image is "shrunk" by setting all voxels that are less than or equal to a chosen threshold h_o for the object, and h_b for the background to zero. In this way, a hollow space is created around the border of the object. To smooth the object, a new border should be placed midway in the hollow space. A computationally convenient way to reach this goal is to compute the distance transform of the hollow space with respect to its complement. Again, positive and negative signs are used so that we can distinguish distance information, depending on whether it comes from object voxels or background voxels. Object voxels are initially set to +1, background voxels to -1 and voxels in the hollow space are set to +infinity. When a voxel in the hollow space has the same distance from both positive and negative voxels, a decision should be taken on the sign to be assigned to the voxel itself. Choosing a negative sign favours smoothing of protrusions, while choosing a positive sign favours removal of concavities. In our case, we favour protrusion flattening and hence choose the negative sign. Once the distance transform of the hollow space is

computed, the image is binarized by setting all voxels with negative value to 0 (i.e., to background), and all voxels with positive value to 1 (i.e., to object).

The two thresholds h_o and h_b are set depending on the size of the largest protrusions or concavities to be smoothed. Equal values for the two thresholds are selected to avoid that the border of the object is altered wherever smoothing is not necessary and to keep the size of the object approximately constant. Any distance transform, i.e., any choice of the local distances w_f, w_e, and w_v is possible. Good results are obtained with w_f =3, w =4, and w_v=5, [7].

A synthetic example showing the effect of smoothing is given in Fig. 1, where white is used for the background, h_o=h_b=3 and w_f =3, w_e=4, and w_v=5. A 2D noisy grey-level image, a), consisting of four square frames, each with different width and with pixels all having uniform grey-level except for a few sparse pixels with either higher or lower grey-level, is converted to the 3D binary image, b), where different grey tones are used only to help readers to associate the regions of the 3D image with the corresponding regions in the 2D image. Thin spikes (and thin elongated concavities) are the 3D visible effect of noise. In the 3D image resulting after smoothing, c), spikes and concavities have been flattened and filled in, respectively. Moreover, 90-degree corners, created in the 3D image due to the sharp change of grey-level in neighbouring frames of the 2D image, have been smoothed. The 3D image is projected to provide the resulting 2D grey-level smoothed image, d).

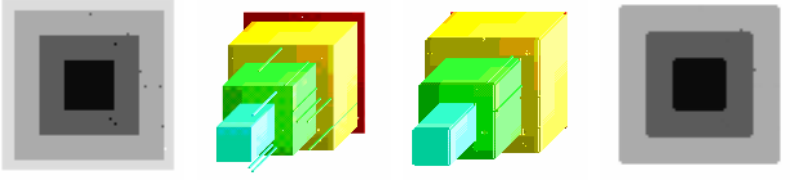


Fig. 1. A noisy 2D grey-level image, a), the converted 3D binary image, b), the smoothed 3D binary image, c), and the smoothed 2D grey-level image, d), obtained by projection

Smoothing can be used in general to improve the image quality. In particular, its use before convex hull computation and skeletonization allows us to obtain smoother covering polyhedra and skeletons with simpler structure. In the following, we use as running example a grey-level digit 6. The effect of smoothing can be seen in Fig. 2.

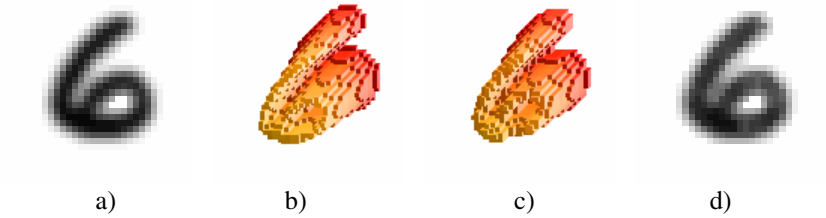


Fig. 2. The original digit 6, a) and b), and the smoothed digit 6, c) and d)

4 Grey-Level Convex Hull Computation

The convex hull of an object is the smallest convex set enclosing the object [8]. For a 2D grey-level image, one could compute the convex hull by taking into account either only geometric information, or also adding grey-level information. What is the best choice depends on the application. Here, we use both geometric and grey-level information and suggest a purely discrete and local approach to compute a sufficiently good approximation of the convex hull of a 2D grey-level object in the corresponding 3D binary image [4]. Our algorithm builds a covering polyhedron, closely fitting the object, by iteratively filling local concavities, [3]. Local concavities are defined as border voxels (i.e., background voxels with at least one face-neighbouring object voxel) with a certain number of neighbouring object voxels. The larger the neighbourhood used to identify local concavities is, the more half-spaces can be used for delimiting the covering polyhedron, and the better the approximation of the convex hull will be. Naturally, the computational cost increases with the size of the adopted neighbourhood. If the 3×3×3 neighbourhood, is used, the computation cost of the algorithm is the smallest possible, but the polyhedron would have at most 26 faces. This is a too rough approximation of the convex hull. On the other hand, resorting to larger neighbourhoods and larger operators, would make the algorithm significantly more heavy.

We have found a good compromise between computational cost and quality of the obtained results. Our method still uses only 3×3×3 operators, but curvature information is derived from the 5×5×5 neighbourhood. This goal is achieved by splitting each iteration in two subiterations. During the first subiteration we count, for each border voxel v, the number of its object face- and edge-neighbours, individually in each of the x-, y-, and z-planes, and store the three resulting sums, S_x, S_y and S_z, as a vector label for v. Then, during the second subiteration, any voxel v with at least one $S_k>4$ (k = x, y, z), or such that one $S_k=4$ and having, in the same plane k, at least one neighbour with $S_k>2$ is defined as being a local concavity and changed from background to object. In this way, a local concavity is filled. By iteratively filling local concavities, the concavity regions are filled globally: The filling process terminates when all border voxels belong to planes oriented along permitted orientations. The resulting covering polyhedron is convex and can have up to 90 faces.

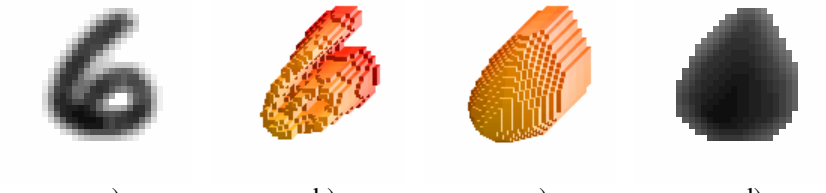


Fig. 3. The 3D covering polyhedron, c), for the smoothed digit 6, a) and b), and the 2D grey-level (approximation of the) convex hull, d), obtained by projection.

The difference between our covering polyhedron and the convex hull is reasonably small, and the covering polyhedron is good enough for most practical purposes. When the binary 3D image is converted to 2D, the resulting convex hull is a grey-level set where both geometric (planar) concavities and grey-level concavities (areas with lower grey-level than their surroundings) of the object are filled.

The performance of the algorithm on the running example can be seen in Fig. 3.

5 Grey-Level Skeletonization

The skeleton of a grey-level object is a thin subset of the object, centered within the regions of the object with locally higher intensity, and having the same topology as the object. Literature on skeletonization is rich for binary images (see, e.g., [9]), but not equally rich for grey-level images (see [10] for an early example). To compute the grey-level skeleton of a 2D object, we use the algorithm introduced in [5] to compute the skeleton of a 3D surface-like object in a binary image. The algorithm is based on the intuitive classification of all voxels belonging to a surface-like object into curve, junction, inner, and edge voxels (see [5] for more details). The advantage of this algorithm is that it uses geometrical information to automatically ascribe to the curve skeleton the voxels that will be its end-points. In fact, it is well known that a critical step in skeletonization algorithms is the correct identification of end points.

We compute the curve-skeleton of the top surface of the 3D object corresponding to the 2D grey-level object [6]. For each (x,y), the voxel with maximal z value belongs to the top surface. However, there is no guarantee that this set is a connected and tunnel-free surface. Thus, we define the top surface of the object as the set consisting of all object voxels in positions (x,y,z) with $z \neq 0$ and having a face-neighbour in the background. The so defined top surface is generally 1-voxel thick, but it may be 2-voxel thick in presence of surface parts forming 90-degree corners. This is typically the case when in the 2D grey-level image a region with constant grey-level g is adjacent to a region whose grey-level is equal to or smaller than g-2 (see, e.g., Fig. 1). These 90-degree corners would be classified as consisting of junction voxels by the classification method used within the skeletonization algorithm, while they actually correspond to an artefact created during the conversion from 2D to 3D. Since the skeletonization algorithm does not remove voxels classified as belonging to junctions, we should remove from the top surface these spurious junctions consisting of corner voxels to obtain a significant skeleton. Indeed, most of these spurious junctions are smoothed by the procedure illustrated in Section 3. Corner voxels still remaining in the top surface are assigned to the background, which is equivalent to decreasing the grey-level g by 1 in the corresponding positions.

The curve skeleton is computed by means of an iterative process. At each iteration, the voxels that are currently classified as curve or edge voxels are identified as border voxels. Among them, only the edge voxels are removed, provided that their removal does not alter topology. In this way, curve voxels (classified as such in the initial top surface or at any successive stage of the process) are automatically preserved so that unwanted shortening of skeleton branches is avoided.

When junction voxels are identified in the initial classification of the top surface, skeletonization is done in two steps to guarantee that the most significant shape in-

formation remains in the resulting skeleton. Voxels classified as junction voxels in the input top surface may be transformed into curve voxels, inner voxels or edge voxels during the removal process. In the first case (curve voxels), they are directly ascribed to the curve skeleton; in the second case (inner voxels), they cannot be considered for removal; finally, in the third case (edge voxels), their removal is possible, but it is done only during the second step of the process. This step starts when all voxels currently classified as edge voxels are those that were classified as junction voxels in the input surface. To this purpose, during the first step, both the current classification and the initial classification are taken into account, iteration after iteration. Voxels, identified as edge voxels according to the current classification, can be removed only if they were not classified as junction voxels in the initial classification. During the second step, only the current classification is used. The resulting set is a nearly-thin skeleton, i.e., an at most 2-voxel thick set.

Final thinning of the curve skeleton can be performed by using standard topology preserving removal operations, e.g., based on the identification of simple voxels [11]. To prevent excessive branch shortening, we split the final thinning into six directional processes (top, back, right, down, front, left), each of which reduces the thickness of the skeleton in the corresponding scanning direction.

Before converting the 3D image to 2D, the 3D curve skeleton is pruned. The classification into edge, curve, inner, and junction voxels done on the top surface is used also to perform pruning without significant loss of shape information. In fact, voxels classified as curve or junction voxels in the top surface are considered as those retaining the most significant shape information and, hence, are marked as significant voxels. Only branches including a small number of significant voxels compared to the total number of voxels in the branch are pruned. Pruning is active only on peripheral branches (i.e., branches delimited by an end point and a branch point). When the structure of the curve skeleton is complex, pruning only peripheral branches may not produce satisfactory results. In fact, internal branches, i.e., branches delimited by two branch points in the initial curve skeleton, may include a large number of non-significant voxels with respect to the total number of voxels. These branches may become peripheral only after removal of adjacent peripheral branches. In such a case, pruning is iterated. Of course, information about the already removed branches is used, to avoid an excessive shortening due to pruning iteration.

a) b)

Fig. 4. The 3D curve skeleton of the smoothed digit 6, a), and the 2D grey-level skeleton, superimposed on the original object, b), obtained by projection

Before applying context dependent pruning, a brute force removal of branches including only a few voxels is performed. These short branches are interpreted as noisy branches even if they include almost only significant voxels. Their presence in the

skeleton would postpone to the second iteration of pruning, or even completely prevent, removal of other longer but still non-significant branches.

At this point, the 3D binary image can be converted to the 2D grey-level skeleton image. In general, final thinning and postprocessing (spurious loop filling and, possibly, pruning) have to be applied on the so obtained 2D skeleton to reduce its business and to achieve a topologically correct result. In fact, 3D curves, separate but close to each other, may originate touching 2D curves, which may produce thickening and spurious loops in the resulting 2D skeleton. The performance of the skeletonization algorithm on the running example is shown in Fig. 4.

6 Conclusion

In this paper, we have shown the results obtained by applying three algorithms, initially intended for 3D binary images, to 2D grey-level images. The 2D grey-level image is initially converted to a 3D binary image, by using the grey-level as the third coordinate. The 3D binary algorithms have been applied to this image and the results have been projected back to 2D to obtain the grey-level resulting images.

The results obtained are promising and we plan to continue along this direction.

References

1. G. Sanniti di Baja, S. Svensson, “Editing 3D binary images using distance transforms”, *Proc. 15th ICPR*, Barcelona, Spain, pp. 1034-1037, 2000.
2. J. Serra, “Image Analysis and Mathematical Morphology” Vol. I, Academic Press, London, 1982
3. G. Borgefors, I. Nyström, G. Sanniti di Baja, “Computing covering polyhedra of non-convex objects”, *Proc. of BMVC94*, York, 275-284, 1994.
4. I. Nyström, G. Borgefors, G. Sanniti di Baja, “2D Grey-level Convex Hull Computation: A Discrete 3D Approach”, in *Image Analysis,* J. Bigun and T.Gustavsson Eds., LNCS 2749, Springer Verlag, Berlin, pp. 763-770, 2003.
5. S. Svensson , I. Nyström, G. Sanniti di Baja, “Curve skeletonization of surface-like objects in 3D images guided by voxel classification”, *Pattern Recognition Letters*, 23/12 pp. 1419-1426, 2002.
6. G. Sanniti di Baja, I. Nyström “2D Grey-level Skeleton Computation: A Discrete 3D Approach”, *Proc. 17 ICPR*, Cambridge, UK, pp. 455-458, 2004.
7. G. Borgefors, “On digital distance transforms in three dimensions”, *Computer Vision Image Understanding* 64/3, pp. 368-376, 1996.
8. F.P. Preparata, M.I. Shamos, “Computational Geometry. An Introduction”, Springer Verlag, New York, 1985.
9. L. Lam, S-W. Lee, C.Y. Suen, “Thinning methodologies. A comprehensive survey”, *IEEE Trans. on PAMI*, 14/9 pp. 869-885, 1992.
10. B.J.H. Verwer, L.J. Van Vliet, P.W. Verbeek, “Binary and grey-value skeletons: Metrics and algorithms”, *IJPRAI,* 7 pp. 1287-1308, 1993.
11. P.K. Saha, B.B. Chaudhuri, “Detection of 3-D simple points for topology preserving transformations with application to thinning”, *IEEE Trans. on PAMI*, 16/10 pp. 1028–1032, 1994.

Slant Correction of Vehicle License Plate Image

Lin Liu, Sanyuan Zhang, Yin Zhang, and Xiuzi Ye

College of Computer Science / State Key Lab of CAD&CG,
Hangzhou, Zhejiang University, China
LiuLin@zju.edu.cn

Abstract. Because of perspective distortion between the camera and the license plate, slanted images commonly appear in License Plate Recognition System (LPR system), and it seriously affects the recognition result. This paper presents two types of image slant, and gives an efficient way to rectify them. For horizontal slant correction, the proposed method is based on connected areas labeling and straight-line fitting. For vertical slant correction, the method is based on rotation experiments with various angles. Practical use in the LPR system shows that this method has a correct rate above 97%.

1 Introduction

1.1 Vehicle License Plate Recognition System

Vehicle License Plate Recognition (LPR) is an important part of Intelligent Traffic System (ITS), and it is widely used in traffic control, traffic management and vehicle security. Commonly LPR system includes the following parts: vehicle image acquisition, license plate image location, image pre-processing and character recognition. The flowchart of our LPR system is described in Fig. 1.

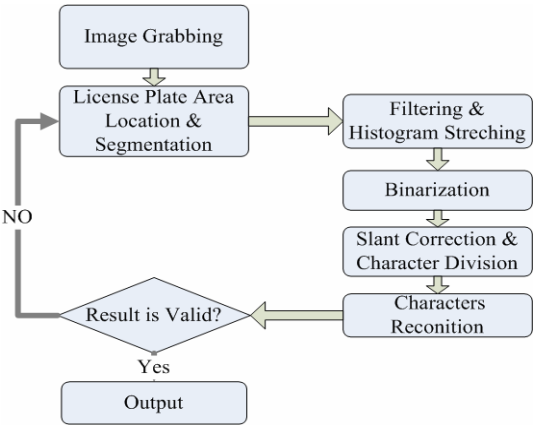


Fig. 1. The flowchart of our License Plate Recognition System

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 237 – 244, 2005.

There is a feedback function in the flowchart. If the recognition result cannot satisfy the syntax requirement (for example, not enough characters), the license plate area is relocated and the license plate is recognized again.

In general, license plate area location decides whether the license plate can be recognized or not, and the image processing module decides final recognition accuracy. Image slant correction is an important part of image processing, which affects the accuracy of character division, even final result.

1.2 Two Types of Image Slant

A segmented license pate image is ideally a rectangle aligning with the vehicle image bottom line, but in practice the license plate image is a parallelogram. This leads to the difficulty of character subdivision. There are two types of image slant, namely horizontal slant and vertical slant, as shown in Fig.2 and Fig.3 respectively. Horizontal image slant often leads to integral slant of license plate image, while vertical slant often leads to characters slant. Two real examples of slanted image with their horizontal projection images are given in Fig.4.

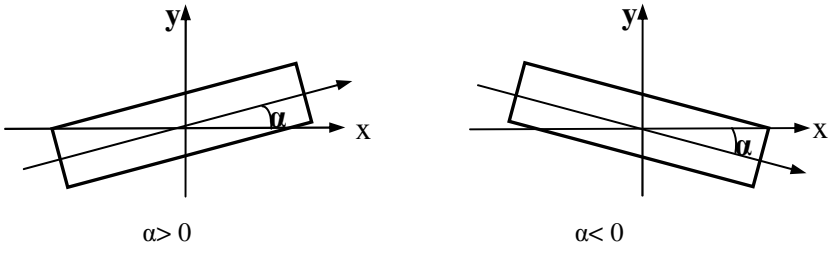


Fig. 2. Two forms of horizontal slant

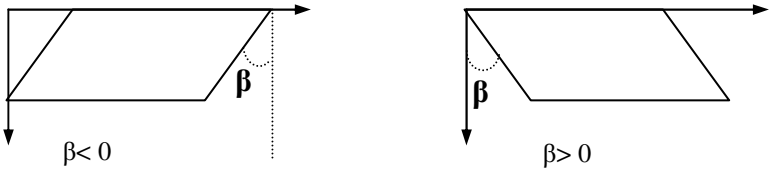


Fig. 3. Two forms of vertical slant

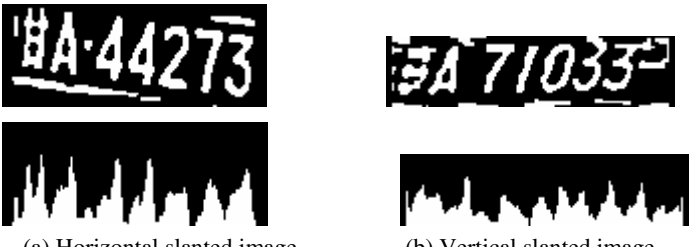


Fig. 4. Two examples of slanted license plate images and their horizontal projections

1.3 Previous Work and Our Method

Slant correction is an important part of LPR system, and it has been intensively studied. A method based on Hough transformation is presented in [1]. It extracts the license plate's top edge and bottom edge using Hough transformation, and the slanted angle is deduced from the edges. A model-matching method is given in [2]. It finds vehicle plate image's four corners by model-matching, then slanted vehicle plate images are transformed to a rectangle area by means of bilinear transformation. A gray value projection method is described in [3][4]. Here the projection image is produced and the license plate area is located by local maximum and minimum projection values. The slant angle is also determined by the projection result. However, it is sometimes difficult to extract the license plate's edges from the background by regular binarization methods (e.g., a white plate on a white vehicle); and due to signal noise and contamination, the plate's edge may be discontinuous; the peak point of Hough transformation is scattered and it is hard to get the acute position of the plate's edge. Therefore, in practice methods based on the plate's edges [1][2] may not get expected results. When the license plate is seriously contaminated, the method in [3] will not work properly.

During our software development, we found that it is relatively easy to separate single character from the whole license plate areas. For horizontal slanted images, if enough character areas are valid, we can use feature points of these character areas to fit a line, and the line can represent the direction of the license plate image. For vertical slanted license plate, its direction is calculated by an empirical method.

2 Horizontal Slant Correction

The flowchart of horizontal slant correction is given in Fig.4. We will ignore image segmentation and image binarization steps in this paper, and focus on other four steps in Fig.5.

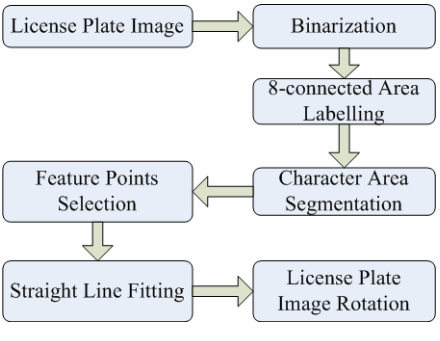


Fig. 5. The flowchart of horizontal slant correction

2.1 8-Connected Areas Labeling

The first step of horizontal slant correction is isolating connected areas from a license plate image, which is implemented by connected areas labeling on binaried license

plate image. After appropriate binarization (here our binarization method is based on OSTU method with a Log filter), an 8-connected area labeling method, rather than a 4-connected one, is adopted to get connected image areas because some image areas are partly discontinuous.

2.2 Character Areas Selection

To separate character areas from noise area or license plate's edges, a connected image area with the following characteristics is considered as a valid character area:

(a) The candidate areas are not too wide, and can be separated from the license plate's horizontal edges.
(b) The candidate areas are not too high.
(c) Most candidate areas have a gap among them.

If more than 3 valid character areas are found, it is enough to decide the slant angle of the license plate.

2.3 Feature Points Selection and Straight-Line Fitting

Top-most points and bottom-most points of each valid character area are selected as feature points, and they are used to fit the top straight-line and bottom straight-line separately using least square algorithm. The angle between top straight line and bottom straight line is calculated to validate the result, if absolute value of the angle is small (commonly < 3°), then the result is considered valid, otherwise the result is discarded. Fig.6 is an example of the result of straight-line fitting (gray line on the image).

Fig. 6. An example of straight-line fitting

2.4 Rotation of License Plate Image

With image rotation, we can get corrected image. Image rotation is not performed on binarized images. This is because direct image rotation on binaried images will bring some noises to the pixels. Fig.7 shows this effect of direct image rotation from Fig.6.

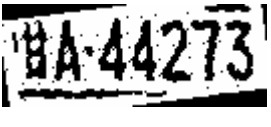


Fig. 7. Direct image rotation's result of Fig.6

Instead, image rotation is performed on the gray image, and final corrected image is obtained from rotated gray image by a binarization step. Fig.8 gives the final result.

(a) Rotated gray image of Fig.6

(b) Final corrected image (Binarized image of (a))

(c) Projection image of (b)

Fig. 8. Image rotation result

Compared with the projection image in Fig.4 (a), we can see that the peak and hollow of Fig.8 (c) is clearer, this will help characters division in Fig.8 (b). In practice, only license plate images slanted with a relatively big angle will be rotated, commonly absolute value of this angle is above 3°.

3 Vertical Slant Correction

Vertical slant images are corrected based on the following facts:

(a) 8-connected areas labeling method can isolate one or more character areas.
(b) The horizontal projection width is smallest when the license plate is not slanted.
(c) Slant angle of the license plate is in a specified range, here we give the range from -20° to $+20^{\circ}$.

Here we give a simple method to compute the rotation angle: at first a character area is selected, then its horizontal projection is calculated with various angles, finally the angle with smallest projection is selected as the license plate's rotation angle.

For the vertical slanted license plate image given in Fig.4, Table 1 gives the rotation result of selected character.

Table 1. Rotation result of selected character in Fig.4

Rotation result								
Rotation angle (°)	-20	-15	-5	0	5	10	15	20

From this table, we can find the rotation angle is –20°. The result of vertical slant correction of Fig.4 is given in Fig.9.

(a) Corrected image (b) The projection of corrected image

Fig. 9. Result of vertical slant correction

Compared with the projection image in Fig.4 (b), we can see that the peak and hollow of Fig.9 (b) is clearer and easier to distinguish. This will help the subdivision of characters in Fig.9 (a).

4 Experiment Results and System Performance

Our method has been embedded into a vehicle license plate system which is very successful commercially. To obtain the performance of this method, we tested more than 1500 license plate images. These plate images are got in various illumination conditions. In Fig.10 and Fig.11, we give more examples of horizontal and vertical slant correction. From Table 2 and Table 3 we can find that there are much more horizontal slanted images than vertical ones. Comparing Table 4 with Table 5, we can find less license plate images are discarded and the recognition ratio is significantly increased after image slant corrections. The system's recognition ratio is above 88.3%, which can fulfill most regular requirements.

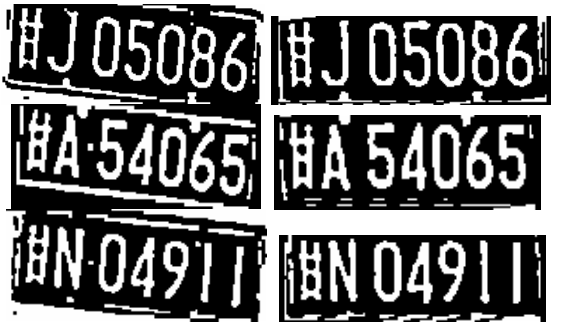


Fig. 10. Examples of horizontal slant corrections

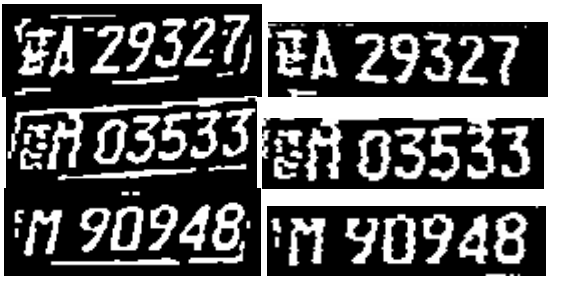


Fig. 11. Examples of vertical slant corrections

Table 2. Horizontal Slant Correction Ratio

Total license plate images	Horizontal slanted images	Corrected images	Correction ratio
1590	688	674	97.9%

Table 3. Vertical Slant Correction Ratio

Total license plate images	Vertical slanted images	Corrected images	Correction ratio
1590	96	88	91.6%

Table 4. Total Recognition Ratio without Slant Correction

Total license plate images	Correctly Recognized images	Discarded images	Correction ratio
1590	1308	185	82.2%

Table 5. Total Recognition Ratio with Slant Correction

Total license plate images	Correctly Recognized images	Discarded images	Correction ratio
1590	1405	42	88.3%

5 Conclusion

Slant correction is an integrant part of vehicle license plate recognition system, and there are commonly two main image slant types: horizontal slant and vertical slant. This paper gives an efficient method to correct slanted images in a commercial license plate recognition system. For horizontal slant correction, our method is based on connected areas labeling and straight-line fitting. For vertical slant correction, our method is based on an empirical method. Practical uses in commercial applications show that our method has a satisfactory correction ratio in all kinds of environmental conditions. Intensive study on the proposed method shows that it is also suitable for slant correction of OCR system.

Acknowledgements

The authors would like to thank the support from the China NSF under grant #602720601, China Ministry of Education under grant # 20030335064, and China Ministry of Science and Technology under grant #2003AA4Z1020, and the project(No G20030433) of the Education Office of Zhejiang Province

References

1. Wen C-Y, Yu C-C, Hun Z-D. A 3-D transformation to improve the legibility of license plate numbers [J], Journal of Foresensic Sciences, 2002, 47(3): 578~585.
2. Hegt H A, Haye R J, Khan N A. A high performance license plate recognition system [A], IEEE international Conference on Systems, Man, and Cybernetics [C]. 1998, 5.4357~4362.
3. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
4. Daiyan,Ma Hongqing,,A high performance license plate recognition system based on the web technique, 2001 IEEE Intelligent Transportaion Systems Conference Proceedings (CA)Oakland,USA-August 25-29

Total Variation-Based Speckle Reduction Using Multi-grid Algorithm for Ultrasound Images

Chen Sheng[1], Yang Xin[1], Yao Liping[2], and Sun Kun[2]

[1] Institution of Image Processing and Pattern Recognition,
Shanghai Jiaotong University, Shanghai, P.R. China
chnshn@hotmail.com, yangxin@sjtu.edu.cn
[2] Shanghai Children's Medical Center, Shanghai Second Medical University,
Shanghai, P.R. China

Abstract. This paper presents an approach for speckle reduction and coherence enhancement of ultrasound images based on total variation (TV) minimization. The proposed method can preserve information associated with resolved object structures while reducing the speckle noise. However, since the equation system deduced by the TV-based method is a strongly nonlinear partial differential equation (PDE) system, the convergence rate is very slow when using standard numerical optimization techniques. So in this paper, we introduce the nonlinear multi-grid algorithm to solve this system. Numerical results indicate that the image can be recovered with satisfied result even contamination of strong noise using the proposed method and the algorithm of nonlinear multi-grid has more efficiency than the conventional numerical techniques such as conjugate gradient (CG).

1 Introduction

The low cost, portability, and real-time image formation make ultrasound imaging an essential tool for medical diagnosis. While ultrasound images often suffer from a special kind of noise called speckle. Speckle significantly degrades the image quality and affects human interpretation of the images as well as the accuracy of computer-assisted diagnostic techniques. So a number of methods have been proposed to address the problem of speckle reduction such as temporal averaging [1], adaptive weighted median filtering (AWMF) [2], adaptive speckle reduction [3], Wiener filtering and wavelet shrinkage [4]. While these approaches could not succeed to balance between speckle suppression and feature preservation due to the complexity of speckle statistics.

In this paper, we describe a speckle reduction technique whereby the ultrasound image is smoothed to suppress the speckle while substantially preserving image components corresponding to resolved object structures. This technique is based on total variation (TV) minimization [5]. Due to its anisotropy, the proposed technique allows coherent structure enhancement while the dynamic smoothing is controlled by the local behavior of the images. In addition, to ensure real-time implementation, we apply nonlinear multi-grid method to solve the nonlinear TV-based minimization problem.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 245 – 252, 2005.

2 Methods

Due to the limited dynamic range of commercial display monitors, ultrasound imaging systems compress the echo signal to fit in the display range. Such compression changes the characteristics of the signal probability density function (PDF). In particular, it affects the high intensity tail of the Rayleigh and Rician PDFs more than the low intensity part. As a result, the speckle noise becomes very close to white Gaussian noise corresponding the uncompressed Rayleigh signal [6]. The statistical properties of speckle noise were studied in [7]. It showed that, when the imaging system has a resolution cell that is small in relation to the spatial detail in the object, and the speckle-degraded image has been sampled coarsely enough that degradation at any pixel can be assumed to be independent of the degradation at all other pixels, coherent speckle noise can be modeled as multiplicative noise. Thus we have:

$$f(x,y) = g(x,y)\eta_m(x,y) + \eta_a(x,y) \tag{1}$$

where: $g(x,y)$ is an unknown piecewise constant two-dimensional function representing the noise-free original image, $f(x,y)$ is the noisy observation of $g(x,y)$, η_m and η_a are multiplicative and additive noise respectively, and x and y are variables of spatial locations that belong to 2-D space of all real numbers, $(x,y) \in R^2$. Since the effect of additive noise is considerably small compared with that of multiplicative noise, Equation (1) can be rewritten as:

$$f(x,y) = g(x,y)\eta_m(x,y) \tag{2}$$

The logarithmic amplification transforms the Equation (2) into the classical additive noise form:

$$\log(f(x,y)) = \log(g(x,y)) + \log(\eta_m(x,y)) \tag{3}$$

The above expression can be rewritten as

$$z(x,y) = u(x,y) + \eta(x,y) \tag{4}$$

At this stage, we can consider $\eta(x,y)$ to be white Gaussian noise and apply a conventional additive noise suppression technique, such as Wiener filtering. It is to find $u(x,y)$ which minimizes the functional:

$$T(u) = \frac{1}{2}\|u - z\|^2 + \alpha J(u) \tag{5}$$

Common choices for J are

$$J(u) = \int u^2 dx \tag{6}$$

Equation (6) often induces blur in images and spurious oscillations when u is discontinuous.

So we consider the nonlinear TV functional:

$$J_{TV}(u) = \int_{\Omega} |\nabla u| dx \tag{7}$$

where: ∇u denotes the gradient of u :

$$\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right)$$

here: u is not required to be continuous.

How ever, the Euclidean norm is not differentiable at zero. To avoid difficulties associated with the non-differentiability, the modification [8]:

$$J_{\beta}(u) = \int_{\Omega} \sqrt{|\nabla u|^2 + \beta^2} dx$$

will be utilized here. Throughout the remainder of this paper, the functional to be minimized is:

$$T(u) = \frac{1}{2} \|u - z\|^2 + \alpha \int_{\Omega} \sqrt{|\nabla u|^2 + \beta^2} dx \tag{8}$$

The Eular-lagrange equation associates with Equation (8) is

$$u + \alpha L(u)u = z, x \in \Omega$$

$$\frac{\partial u}{\partial n} = 0, x \in \partial\Omega \tag{9}$$

where: $L(u)$ is differential operator whose action on u is given by:

$$L(u)u = -\nabla \cdot \left(\frac{1}{\sqrt{|\nabla u|^2 + \beta^2}} \nabla u \right) \tag{10}$$

It is an elliptic nonlinear PDE. From Equation (10), we can see that the smoothing decreases as the gradient strength increases and the smoothing is stopped across edges.

3 Numerical Solutions

There are many standard numerical optimization techniques such as CG method. However, these standard methods tend to perform poorly on TV minimization problems. The underlying difficulty is that the local quadratic approximation that is the basis for standard CG method is not a good model for the nonlinear TV-based minimization. So, in this paper we adopt the nonlinear multi-grid method to deal with this problem.

Multi-grid method was firstly introduced by Brandt [9]. Unlike the conventional methods, the multi-grid algorithm can solve nonlinear elliptic PDE with non-constant coefficients with hardly any loss in efficiency. In addition, no nonlinear equations need be solved, except on the coarsest grid.

Suppose we discrete the nonlinear elliptic PDE of Equation (9) on a uniform grid with mesh size h:

$$T_h(u_h) = z_h \tag{11}$$

where: $T_h(u_h)$ denote $u_h + \alpha L_h(u_h)u_h$.

Let $\tilde{u_h}$ denote some approximate solution and u_h denote the exact solution to Equation (11). Then the correction is:

$$v_h = u_h - \tilde{u_h}$$

The residual is:

$$T_h(\tilde{u_h} + v_h) - T_h(\tilde{u_h}) = f_h - T_h(\tilde{u_h}) = -d_h \tag{12}$$

Now, we form the appropriate approximation T_H of T_h on a coarser grid with mesh size H (we will always take $H = 2h$). The residual equation is now approximated by:

$$T_H(u_H) - T_H(\tilde{u_H}) = -d_H \tag{13}$$

Since T_H has smaller dimension, this equation will be easier to solve. To define $\tilde{u_H}$ and d_H on the coarse grid, we need a restriction operator R that restricts $\tilde{u_h}$ and d_h to the coarse grid. That is, we solve:

$$T_H(u_H) = T_H(R\tilde{u_h}) - Rd_h \tag{14}$$

on the coarse gird. Then the coarse-grid correction is:

$$\tilde{v_H} = u_H - R\tilde{u_h}$$

Once we have a solution $\tilde{v_H}$ on the coarse gird, we need a prolongation operator P that interpolates the correction to the fine gird:

$$\tilde{v_h} = P\tilde{v_H}$$

So we have:

$$\tilde{u_h}^{new} = \tilde{u_h} + P\tilde{v_H} \tag{15}$$

It is the two-grid algorithm and can be easily extended to multi-grid.

The symbol of P is found by considering v_H to be 1 at some mesh point (x, y), zero elsewhere, and then asking for the values of Pv_H. The most popular prolongation operator is simple bilinear interpolation. It gives nonzero values at the 9 points $(x, y), (x+h, y), ..., (x-h, y-h)$ and its symbol is:

$$\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \tag{16}$$

The symbol of R is defined by considering v_h to be defined everywhere on the fine grid, and then asking what is Rv_h at (x, y) as a linear combination of these values. The choice for R is the adjoint operator to P. So that the symbol of R is:

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \tag{17}$$

We can see that there are two complementary viewpoints for the relation between coarse and fine grids:

1. Coarse grids are used to accelerate the convergence of the smooth components of the fine-grid residuals.
2. Fine girds are used to compute correction terms to the coarse-grid equations, yielding fine-grid accuracy on the coarse girds.

At the coarsest-grid, we have one remaining task before implementing our nonlinear multi-grid algorithm: choosing a nonlinear relaxation scheme. Our first choice is the nonlinear Gauss-Seidel scheme. If the discretized Equation (11) is written with some choice of ordering as:

$$T_i(u_1,...,u_N) = z_i, i = 1,...,N \tag{18}$$

Then the nonlinear Gauss-Seidel schemes solves:

$$T_i(u_1,...,u_{i-1},u_i^{new},u_{i+1},...,u_N) = z_i \tag{19}$$

for u_i^{new}. Often Equation is linear in u_i^{new}, since the nonlinear terms are discretized by means of its neighbors. If this is not the case, we replace Equation (19) by one step of a Newton iteration:

$$u_i^{new} = u_i^{old} - \frac{T_i(u_i^{old}) - z_i}{\partial T_i(u_i^{old})/\partial u_i} \tag{20}$$

4 Experiments

In this paper, we use the ultrasound images (256×256) to test our method and the algorithm has been implemented using an Intel Pentium IV 1Ghz with 128M RAM, under the Visual C++ 6.0 environment.

We compared the results of our approach with other speckle reduction techniques including AWMF, Wiener filtering. The quality measurements of mean-square error (MSE) and signal-to-MSE ratio (SMSE) were computed and shown in Table.1.
The MSE is defined as:

$$MSE = \frac{1}{K}\sum_{i=1}^{K}\left(\tilde{g}_i - g_i\right)^2$$

where: $\tilde{g}$ is the denoised image, g is the original image and K is image size.

$$SMSE = 10\log_{10}\left(\sum_{i=1}^{K} g_i^{\,2} \Big/ \sum_{i=1}^{K}\left(\tilde{g}_i - g_i\right)^2\right)$$

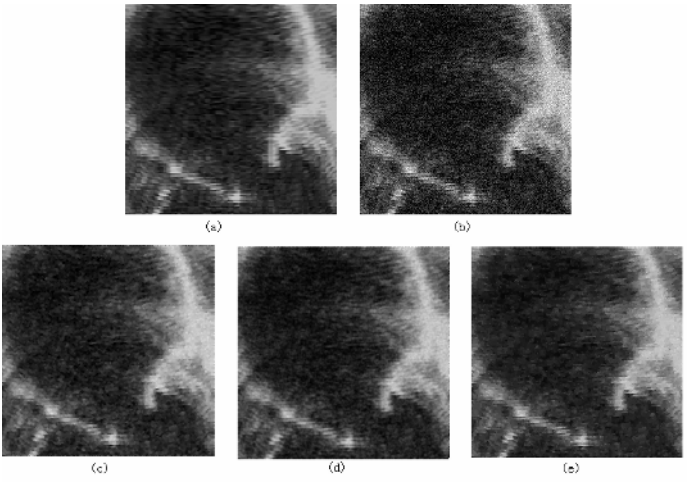


Fig. 1. Results of various speckle reduction methods. (a) Original mitral valve ultrasound image. (b) Image degraded with simulated speckle noise. (c) AWMF. (d) Wiener filtering. (e) Proposed method.

Table 1. Quality measures obtained by three denoised methods tested on speckled mitral valve ultrasound image at various noise levels

Method	MSE	SMSE	MSE	SMSE	MSE	SMSE
Without Filtering	296	13.7	125	17.4	557	10.9
AWMF	134	17.2	79	19.5	215	15.1
Wiener Filtering	80	19.3	53	21.2	150	16.6
Proposed method	71	20.0	40	22.4	122	17.5

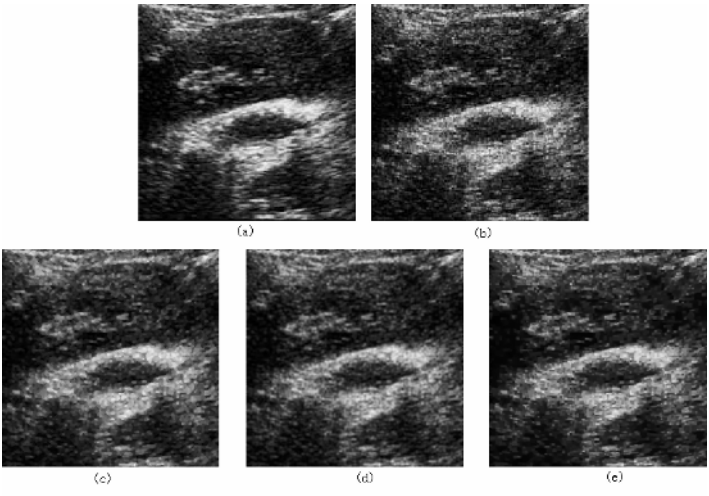


Fig. 2. Results of various speckle reduction methods. (a) Original kidney ultrasound image. (b) Image degraded with simulated speckle noise. (c) AWMF. (d) Wiener filtering. (e) Proposed method.

Table 2. Time consuming comparison for solving the nonlinear TV-based equation

Image size	256×256	180×180	128×128	80×80	64×64	40×40
CG	15.42s	8.20s	3.7s	1.42s	0.63s	0.33s
Multi-Grid	5.77s	2.82s	1.32s	0.55s	0.33s	0.23s

The denoised images are shown in Fig.1 and Fig.2 for visual comparison. The results show that the TV-based speckle reduction method performs better than the AWMF, as well as the Wiener filter. In addition, from Table.2 we can see that the nonlinear multi-grid algorithm is more suitable for the nonlinear TV-based minimization problem than the CG method.

5 Conclusions

In this paper, a new nonlinear method for speckle suppression in ultrasound images is presented. The main innovation is the use of TV regulation to reduce ultrasound speckle while preserving the appearance of structured regions and organ surfaces. By applying the multi-grid nonlinear algorithm, the technique has the advantage of speed of computation and has a large potential in real-time ultrasound imaging enhancement.

Acknowledgments

This work was partially supported by National Science Research Program of China (No. 2004BA714802) and Shanghai Science and Technology Development Foundation (034119820).

References

1. C.B. Burkhardt, "Speckle in ultrasound B-mode scans," IEEE Trans. Sonics Ultrasonic, 1978(25)(1), 1-6.
2. T. Loupas, W.N. Mcdicken and P.L.Allan, "An adaptive weighted median filter for speckle suppression in medical ultrasonic images," IEEE Trans. Circuits System, 1989(36), 129-135.
3. J.C. Bamber and C. Daft, "Adaptive filtering for reduction of speckle in ultrasound pulse-echo images," Ultrasonics, 1986, 41-44.
4. S. Gupta, R.C. Chauhan, S.C. Sexana, "Wavelet-based statistical approach for speckle reduction in medical ultrasound images," IEE Med.Biol.Eng.Comput. 2004(42),189-192.
5. L. Rudin, S. Osher and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Physica D, 1992(60), 259-268.
6. V. Dutt, "Statistical analysis of ultrasound echo envelope," Ph.D. dissertation, Mayo Graduate School, Rochester, MN, 1995.
7. J.W. Goodman, "Some fundamental properties of speckle", Journal of Opt. Soc. Am, 1976(66), 1145-1150.
8. Curtis. RV, Mary EO, "Fast, robust total variation-based reconstruction of noisy, blurred images", IEEE Transactions on Image processing, 1998 7(6), 813-824.
9. Brandt. A, "In Multi-grid Methods", Springer Lecture Notes in Mathematics, 1982, No. 960.

Contour Co-occurrence Matrix – A Novel Statistical Shape Descriptor

Rami Rautkorpi and Jukka Iivarinen

Helsinki University of Technology,
Laboratory of Computer and Information Science,
P.O. Box 5400, FIN-02015 HUT, Finland
{rami.rautkorpi, jukka.iivarinen}@hut.fi

Abstract. In this paper a novel statistical shape feature called the Contour Co-occurrence Matrix (CCM) is proposed for image classification and retrieval. The CCM indicates the joint probability of contour directions in a chain code representation of an object's contour. Comparisons are conducted between different versions of the CCM and several other shape descriptors from e.g. the MPEG-7 standard. Experiments are run with two defect image databases. The results show that the CCM can efficiently represent and classify the difficult, irregular shapes that different defects possess.

1 Introduction

There are lots of different features available that are used in image classification and retrieval. The most common ones are color, texture and shape features [1]. Shape features can be divided into two main categories [2]: syntactical, using structural descriptions suitable for regular shapes such as man-made objects, and statistical, which is more suitable for irregular, naturally occuring shapes. Statistical features can be extracted efficiently using histogram techniques, which are popular due to their simplicity as well as their good performance.

In this paper a novel statistical shape feature called the Contour Co-occurrence Matrix (CCM) is proposed. The CCM indicates the joint probability of contour directions in a chain code representation of an object's contour. Different versions of the CCM are experimented with and comparisons are made between them and several other shape descriptors from e.g. the MPEG-7 standard. The classification performance is tested with two defect image databases. Some earlier work with these databases and the MPEG-7 features are found e.g. in [3,4].

2 Contour Co-occurrence Matrix (CCM)

The Contour Co-occurrence Matrix (CCM) contains second-order statistics on the directionality of the contours of objects in an image. It resembles the gray level co-occurrence matrix (GLCM) [5], but instead of a two-dimensional image, the co-occurrence information is calculated from the Freeman chain code [6] of the contour of an object. In this regard, it is related to the Chain Code Histogram (CCH) [7] which is the first-order counterpart of CCM.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 253–260, 2005.

2.1 Feature Extraction

The first step in calculating the CCM of an object is to generate the chain code of its contour. The starting point of the contour is not stored, so the resulting feature descriptor is translation invariant.

The co-occurrence matrix is then formed from the pairs of links separated by a given displacement. Let A be a chain of length n and let d be a displacement, i.e. the difference between two chain link indices (not the distance, i.e. the absolute value of the difference). Then the contour co-occurrence matrix $\mathbf{H}^{CCM}$ is defined as a matrix, where the (i, j)th element is the number of instances of a link with value i separated from a link with value j by the displacement d,

$$H_{ij}^{CCM} = \#\{k \mid a_k = i,\ a_{k+d\,(mod\ n)} = j\}, \tag{1}$$

where # is the number of elements in the set and k runs through the values $0, 1, \ldots n-1$. Because the chain is derived from a closed contour, the index k and displacement d are summed modulo n, so that the displacement wraps around the chain's arbitrary starting point. Since the chain code is octal, the size of the CCM is 8×8.

Implementing rotation invariance is problematic. The contour direction is quantized into eight values, and certain rotation angles result in predictable transformations of the CCM. A rotation of 90° shifts the elements of the matrix by two steps. Rotations of 45° result in a similar shifting effect, but due to the rectangular grid, there can be a significant change in the distribution of edge pixels, an effect similar to quantization noise. Invariance with respect to these rotations can be achieved by matching shifted versions of the matrix, but the differing lengths of chain links in different directions have to be taken into account by normalizing the elements of the CCM to the lengths of the respective link directions.

Multiple displacements can be used in order to obtain information about the contour at different scales, thereby improving the performance of the descriptor. The resulting matrices can be either summed or concatenated to produce the final feature descriptor.

Two basic variations of the CCM may be considered, based on whether the displacement d is constant over all examined contours (let this be called the CCM1), or dependent on the length of the chain, i.e. $d = cn$, where c is a real number in the range $[0, 1[$ (let this be called the CCM2). If the sum of the CCM's elements is normalized to unity, the matrix represents the joint probability of the link values i and j occurring at link indices with the difference d. Thus normalized, the CCM2 becomes scale invariant.

A matrix that has been normalized to unit sum can be interpreted as a probability distribution, from which a set of features can be calculated. These features were originally proposed by Haralick [5] for use with the GLCM, and additional features were introduced by Conners and Harlow [8]. This way the dimensionality of the feature vector can be reduced, which makes computations such as distance calculations more efficient.

2.2 Example CCMs

Some examples of the CCM, calculated as described later in Section 4.1, are presented in Figure 1. The matrices are presented as bitmaps, with intensity representing bin values. For clarity, the values have been normalized so that the highest bin value in a matrix is shown as white. These images show how the CCM captures some general properties of an object. The indices 2 and 6 represent the vertical directions. For a vertical stripe, the matrix contents are concentrated on the intersections of these indices, representing the relationships of points on the same side of the contour, at $(2, 2)$ and $(6, 6)$, and points on opposite sides, at $(2, 6)$ and $(6, 2)$. For a cluster of somewhat round spots with some distinctly vertical features, the contents are spread more loosely around the same elements. For a slightly vertically elongated, irregular spot, some of the same structure is visible, but the contents are more spread out and slightly shifted away from the indices that would represent a regular, vertical shape.

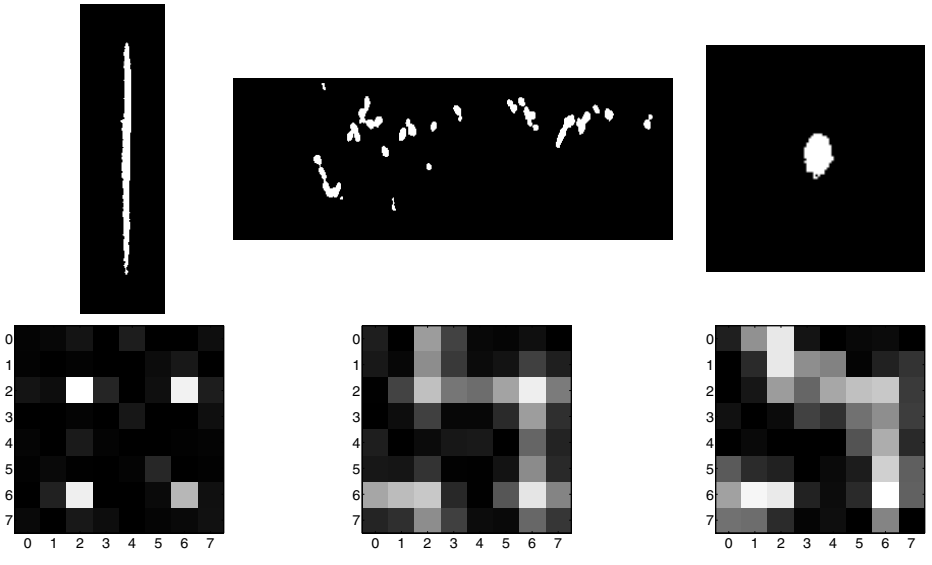


Fig. 1. Contour Co-occurrence Matrices for three images from the metal database

3 Other Shape Descriptors

Other shape descriptors considered in this paper are taken from the MPEG-7 standard, formally named "Multimedia Content Description Interface" [9]. These descriptors were selected for several reasons. They are well standardized descriptors that are used in searching, identifying, filtering and browsing images or video in various applications. In addition to MPEG-7 shape features we also tested three other shape descriptors that we have used previously for defect image classification and retrieval.

Edge Histogram (EH). (MPEG-7) calculates the amount of vertical, horizontal, 45 degree, 135 degree and non-directional edges in 16 sub-images of the picture, resulting in a total of 80 histogram bins.

Simple Edge Histogram (SEH). is similar to its MPEG-7 counterpart, but instead of dividing an image into several sub-images, it is calculated for the whole image.

Contour-based Shape (CBS). (MPEG-7) consists of a set of peak coordinates derived from a Curvature Scale Space (CSS) representation of a contour, and the eccentricities and circularities of the contour and its convex prototype, which is created by repeatedly low-pass filtering the contour.

Region-based Shape (RBS). (MPEG-7) utilizes a set of 35 Angular Radial Transform (ART) coefficients that are calculated within a disk centered at the center of the image's Y channel.

Simple Shape Descriptor (SSD). [10] consists of several simple descriptors calculated from an object's contour. The descriptors are convexity, principal axis ratio, compactness, circular variance, elliptic variance, and angle.

Chain Code Histogram (CCH). [7] is an 8-dimensional histogram calculated from the Freeman chain code of a contour. It is the first-order equivalent of the CCM.

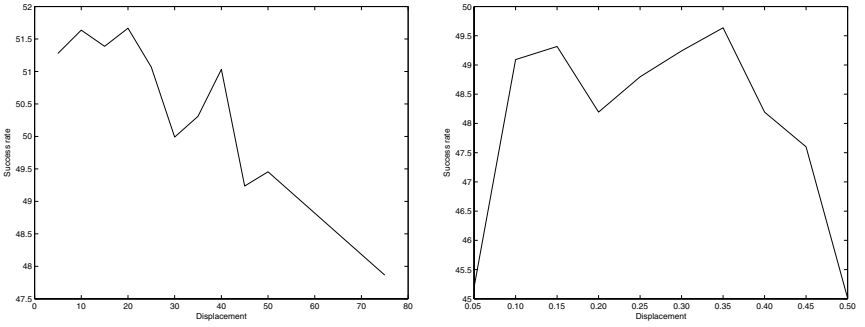
4 Experiments

Experiments were carried out with two image databases containing defect images, one from a metal web inspection system and the other from a paper web inspection system. All images are grayscale images, supplied with binary mask images containing segmentation information, from which the contours of the objects were extracted. The images have different kinds of defects and their sizes vary according to the size of a defect. Classification of defects is based on the cause and type of a defect, and different classes can therefore contain images that are visually dissimilar in many aspects. The paper defect database has 1204 images. They are preclassified into 14 different classes with between 63 and 103 images in all of the classes but one which has only 27 images. The metal defect database has 2004 images. They are preclassified into 14 different classes, with each class containing from 101 up to 165 images. The databases were provided by ABB Oy. More information on these databases can be found e.g. in [3,4].

Classification performance was tested with K-Nearest Neighbor leave-one-out cross-validation, using $K = 5$ and the Euclidean distance measure.

4.1 Displacement Selection

Some initial experiments were made to determine good displacement values. For the CCM1, the range of possible displacements is clearly too large to be exhaustively searched. For a given contour, the CCM1 is periodical with respect



(a) CCM1, metal database (b) CCM2, metal database

Fig. 2. Classification success rates (%) using different displacements

to a displacement. While the period for a single contour is its length n, the period for a set of contours is the least common multiple of the lengths of all the contours. Figure 2(a) shows that for the metal database good results are obtained with displacements in the range from 10 to 40, from which 10 and 20 were chosen. For the paper database a good range is from 20 to 80, and the displacements 20 and 40 were chosen. These values depend on the dataset in a very complex way. In contours of different lengths, a specific displacement value represents a very different relationship between points. The good values for displacement emerge from the entirety of the dataset.

With the CCM2 it is easier to select displacements that can be expected to give sufficiently good results, since the displacement parameter has the range [0 1[. Disregarding rounding effects, the CCM calculated using a relative displacement of c is the transpose of the CCM calculated using a relative displacement of $1 - c$, and therefore only the range $[0, 0.5]$ needs to be examined. Figure 2(b) shows the classification success rates using relative displacements from 0.05 to 0.50 at intervals of 0.05. Based on these results, the relative displacements 0.10, 0.20, 0.30, and 0.40 were chosen, and the matrices were summed together to form the feature vector. Since the displacement is relative to chain length, these choices can be expected to give good results in other databases as well.

4.2 Comparison with Different CCMs

Classification results using the descriptors CCM1 and CCM2 as developed in Section 2.1 are presented in Table 4.2. Although the CCM1 performed better, it also required more care in selecting the displacements. If optimizing the selection of displacements is not possible, e.g. the database grows during use, and the initial set is not representative of the actual pool of data being used, then the CCM2 is probably more reliable, due to the use of relative displacements. Here we assume that the training set is representative of the data, so using the CCM1 with optimized displacements gives a slight advantage. In the remaining experiments only the CCM1 will be used, and will be referred to simply as the CCM.

Table 1. Comparison between the CCM1 and the CCM2

	Classification success rates (%)			
	CCM1 unnorm.	CCM1 norm.	CCM2 unnorm.	CCM2 norm.
Metal	53	49	51	47
Paper	56	58	55	54

4.3 Comparison with Other Shape Descriptors

A comparison was made between the other shape descriptors and the unnormalized CCM, the normalized version and the extracted Haralick features. The 14 features suggested by Haralick in [5] and cluster shade and cluster prominence, added by Conners and Harlow in [8], were considered. The set of features was pruned with a greedy algorithm that eliminated on each iteration the feature that would result in the smallest decrease in the classification rate. The selected features are listed in Table 2. Since the values of the features have very different ranges, the feature vector was whitened using PCA, resulting in a vector where each component has zero mean and unit variance.

Table 2. The sets of features calculated from the CCM and used in classification

Metal	Paper
Difference entropy	Inverse difference moment
Information measures of correlation 1	Entropy
Cluster shade	Information measures of correlation 2
Cluster prominence	Cluster prominence

The CCM results are compared with those obtained with six other shape descriptors: the Edge Histogram (EH), the Simple Edge Histogram (SEH), the Contour-based Shape (CBS), the Region-based Shape (RBS), the Simple Shape Descriptors (SSD), and the Chain Code Histogram (CCH). The results are shown in Tables 3 and 4.

In the metal database the best descriptor was the unnormalized CCM, which scored 4% better than both the normalized CCM and the EH. However, the advantage over the SEH was 9%. Haralick features scored 11% lower than the unnormalized CCM. The CBS and the RBS were clearly the weakest, with the CCH and the SSD falling in-between.

In the paper database the best descriptor was the EH, scoring 2% better than the normalized CCM, which in turn was 2% better than the normalized CCM. However, the SEH was considerably worse, scoring 18% lower than the normalized CCM. This shows that in the paper database dividing the images into segments gives a great advantage. In the metal database the difference was much smaller. Haralick features scored 6% lower than the normalized CCM, the same as the SSD, slightly better than the CBS and the RBS. The CCH was the worst one.

Table 3. KNN classification results of the metal database

	Classification success rates (%)														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	avg
CCM unnorm.	25	51	54	41	59	69	56	33	46	50	82	18	52	99	53
CCM norm.	32	25	59	21	27	56	46	48	67	38	91	21	56	99	49
CCM Haralick	25	32	15	32	41	60	47	38	37	30	73	12	46	96	42
EH	37	45	27	30	62	63	54	26	55	61	61	21	33	91	49
SEH	30	15	15	22	29	71	59	14	60	24	87	33	42	97	44
SSD	29	45	8	33	42	59	55	40	25	46	44	16	43	96	42
CCH	32	23	9	22	21	46	41	36	34	20	63	18	36	95	36
CBS	15	38	14	29	42	38	32	11	13	51	42	7	26	65	31
RBS	16	13	13	9	30	37	25	8	17	14	16	7	11	65	20

Table 4. KNN classification results of the paper database

	Classification success rates (%)														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	avg
CCM norm.	45	68	43	95	54	28	31	71	82	67	42	11	46	94	58
CCM unnorm.	31	58	44	93	55	19	38	80	82	65	39	19	50	92	56
CCM Haralick	39	55	38	85	48	23	33	59	65	71	33	10	53	89	52
EH	41	23	72	93	58	31	50	74	54	86	30	8	71	93	60
SSD	53	41	40	84	35	23	42	77	70	74	19	6	33	89	52
CBS	48	46	31	69	28	28	46	76	71	74	10	4	19	89	49
RBS	57	26	50	79	17	33	44	43	27	71	17	1	37	86	46
CCH	33	30	41	69	29	16	29	56	49	57	26	1	18	68	40
SEH	31	28	31	93	25	31	30	34	16	77	11	0	25	75	40

5 Discussion

The CCM was developed for use as a part of a feature set in a surface inspection application. The feature set is used in PicSOM [11], a content-based image retrieval system developed in the Laboratory of Computer and Information Science at Helsinki University of Technology. The set contains three MPEG-7 feature descriptors: the Color Structure (CS) for color, the Homogeneous Texture (HT) for texture, and the Edge Histogram (EH) for shape description. The Simple Shape Descriptors (SSD), representing a different approach to shape description, is also included in the set. The dominant feature in this set is the texture feature, while the shape features contribute the least to the retrieval performance. Nevertheless, the CCM has been found to work as well or slightly better than the other shape descriptors in experiments with different feature sets in both KNN experiments and the PicSOM system.

6 Conclusions

In this paper a novel statistical shape feature called the Contour Co-occurrence Matrix (CCM) was presented. The classification performance was tested and

compared with several other shape features using two defect image databases. The results in all cases show the CCM to be quite efficient.

The length of the CCM feature vector can be decreased by calculating a set of Haralick features from the matrix. It is possible to keep the decrease in performance quite low by selecting the features individually for each dataset.

Acknowledgments. The financial supports of the Technology Development Centre of Finland (TEKES's grant 40102/04) and our industrial partner ABB Oy (J. Rauhamaa) are gratefully acknowledged.

References

1. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. d Chapman & Hall Computing, London (1993)
2. Marshall, S.: Review of shape coding techniques. Image and Vision Computing **7** (1989) 281–294
3. Pakkanen, J., Ilvesmki, A., Iivarinen, J.: Defect image classification and retrieval with MPEG-7 descriptors. In Bigun, J., Gustavsson, T., eds.: Proceedings of the 13th Scandinavian Conference on Image Analysis. LNCS 2749, Gteborg, Sweden, Springer-Verlag (2003) 349–355
4. Iivarinen, J., Rautkorpi, R., Pakkanen, J., Rauhamaa, J.: Content-based retrieval of surface defect images with PicSOM. International Journal of Fuzzy Systems **6** (2004) 160–167
5. Haralick, R., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics **SMC-3** (1973) 610–621
6. Freeman, H.: Computer processing of line-drawing images. Computing Surveys **6** (1974) 57–97
7. Iivarinen, J., Visa, A.: Shape recognition of irregular objects. In Casasent, D.P., ed.: Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling. Proc. SPIE 2904 (1996) 25–32
8. Conners, R., Harlow, C.: Toward a structural textural analyser based on statistical methods. Computer Graphics and Image Processing **12** (1980) 224–256
9. Manjunath, B.S., Salembier, P., Sikora, T., eds.: Introduction to MPEG-7: Multimedia Content Description Interface. John Wiley & Sons Ltd. (2002)
10. Iivarinen, J., Visa, A.: An adaptive texture and shape based defect classification. In: Proceedings of the 14th International Conference on Pattern Recognition. Volume I., Brisbane, Australia (1998) 117–122
11. Laaksonen, J., Koskela, M., Laakso, S., Oja, E.: Self-organising maps as a relevance feedback technique in content-based image retrieval. Pattern Analysis and Applications **4** (2001) 140–152

Kernel Based Symmetry Measure

Bertrand Zavidovique[2] and Vito Di Gesù[1]

[1] IEF, University of Paris XI, ORSAY, France
[2] DMA, Università di Palermo, Italy

Abstract. In this paper we concentrate on a measure of symmetry. Given a transform S, the kernel SK of a pattern is defined as the maximal included symmetric sub-set of this pattern. A first algorithm is outlined to exhibit this kernel. The maximum being taken over all directions, the problem arises to know which center to use. Then the optimal direction triggers the shift problem too. As for the measure we propose to compute a modified difference between respective surfaces of a pattern and its kernel. A series of experiments supports actual algorithm validation.

1 Introduction

This paper deals with capturing approximate symmetry from pictures, wherever it could show in them.

Symmetry is a prevalent perceptive feature for humans. From the survey by Zabrodsky [ZA1], we retain results corroborating our own findings for machines: a) saliency of vertical symmetry associated with a mental rotation: detected symmetry is in the order vertical, horizontal, bent and then rotational; b) symmetry of parts near the axis contribute more than symmetry of further parts near edges, themselves more critical than regions in between.

Symmetry is important in machine vision too as proven by an extensive literature. See [OM1] for a recent quite interesting survey. Models of symmetry suffer three major drawbacks: $\mathbf{d_1}$ edges mainly support symmetry detection; $\mathbf{d_2}$ perfect symmetry is targeted; $\mathbf{d_3}$ the center of mass is assumed to be the focus of attention.

Similar difficulties were long solved for other features as edges, regions or motion in actually measuring the phenomenon − edginess , uniformity, set-direction − to make decisions from the measure rather than from a strict distance. We addressed $\mathbf{d_1}$ in a previous work [DG3] leading to define iterative transforms as the IOT that better accounts for the inner object. In the present paper we tackle the three $\mathbf{d_i}$-difficulties together. After a short bibliography in section 2 reviewing elementary tools, we first introduce the notion of a *kernel* in section 3. It stems logically from IOT through a gauge classical in functional analysis : maximal included (resp. minimal including) set with same property. In section 4, a series of experiments on both binary and grey scaled pictures allow to check proposed techniques, their sensitivity to the center position and the validity of the degree of symmetry. Discussion and further comments conclude the paper.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 261–268, 2005.

2 State of the Art

The *Symmetry Axial Transform (SAT)* [BL1] can be considered seminal for symmetry detection starting from borders of an object. Exhibiting centers of maximal circles, *SAT* retrieves only axes of symmetry already included in the medial axis. Some limitations are solved by the *Smoothed Local Symmetry* [BR1]. Global symmetry, if any, is retrieved from the local curvature of contours, through the locus of mid-edge-point pairs. Likewise, Sewisy [SE1] couples the Hough transform with geometric symmetry to exhibit candidate ellipse centers. In [FU1] polygonal approximations of contours are broken into elements (e.g. Delaunay/Voronoi method) of which axial features are pieced together. Gray level symmetry was firstly investigated in [MA1], where the descriptor is based on cross correlation. In [DG1], symmetries stem from evaluating the *axial moment* of a body around its center of gravity. This descriptor has been applied at a local level to define the *Discrete Symmetry Transform* (*DST*). In [MN1], local reflectional symmetry is computed in convolving with the first and second derivative of Gaussian's. Both a "measure" of symmetry and an axis orientation are provided at each point. Shen [SH1] or DuBuff [BI1] use complex moments associated with Fourier or Gabor transforms for image approximation.

In [KI1], authors introduce several descriptors from Marola's one, further extended to finite supports and varying scales based on Radon and Fourier transforms. A *global optimization* approach is implemented by a probabilistic genetic algorithm to speedup computations. Along the same line, Shen and al. [SH1] detect symmetry in seeking out the lack of it. The asymmetric part of their measure (energy) is null for a set of pixels invariant through horizontal reflection, hence the minimization. In [CR1] (see also [DG2]), a vector potential is similarly constructed from the gradient field. Edge and symmetry lines are extracted through a topographical analysis of the vector field (i.e. curl of the vector potential) at various scales. Symmetry axes are lines where the curl of the vector vanishes and edges are where the divergence of the potential vanishes. Yeshurun and al. [RE1] build on the Blum-Asada vein: they quantify a potential for every pixel to be center of symmetry, based on pairs of edge points tentatively symmetric from their gradient vectors. A symmetry degree is assigned to every pair within a pixel neighborhood and a weighted combination of these makes the potential, whose local maxima provide a measure depending on both intensity and shape.

Some methods provide symmetry descriptors for measures to be computed, others aim at straight symmetry measures. The difference is obvious in comparing for instance Cross's and Yeshurun's. Finally, preceding works show that: 1- introducing true measures helps building more abstract versions of distances, more suitable for approximate comparison of objects; 2- sets which measures apply on may be "sets of pixels or vectors" (shapes) or "sets of patterns" (class transforms): in either case *set operations* , as Minkowski's ones, are worth considered. They do not limit to contours and bridge logic with geometry.

Before to explain why and how we put these ingredients together, let us conclude by pointing out two more works that fit the algorithmic line above and are

the closest to ours. It makes clear the main contributions of this paper respective to previous work. R. Owens [OM2] searches explicitly for a measure to indicate approximate bilateral symmetry of an object. But she defines tentative symmetries from the principal axes of inertia, whence the centroid again. Although it is not mentioned, her measure based on the sum of absolute differences of grey levels in symmetric pairs amounts to a slightly modified L_1-difference between the object and a maximal-for-inclusion symmetric version of it in the given direction. Kazhdan et al. [KA1] use explicitly the same idea of a difference (L_2 in their case) between the image and its closest symmetric version. They need a measure that integrates all reflective invariance about a bundle of straight lines (or planes in 3-D) and a center is necessary to this representation.

3 The New Symmetry Measure

In previous papers [DG3] we defined the *IOT* that is a map product of iterated morphological erosion and symmetry detection.

Definition 1. The *Symmetry Transform*, S, on a continuous object $X \subset R^2$ is:

$$S_\alpha(\mathtt{X}) = \int_X m(x) \times \rho^2(x, r(\alpha))dx \quad for \quad \alpha \in [0, \pi[\qquad (1)$$

where, $r(\alpha)$ is the straight line with slope α passing through the center of gravity of the object X, $m(x)$ is the mass of the object in $x \in X$, and ρ is a distance function of x from the straight line.$\diamondsuit$

Each value of S represents the axial moment of X.

Definition 2. The *Iterated Object Transform*, *IOT*, is given by:

$$IOT_{\alpha,1}(X) = S_\alpha(X) \;\; ; \quad IOT_{\alpha,n}(X) = S_\alpha\left[(\mathtt{E})^{n-1}(X)\right] \qquad for \quad n > 1 \qquad (2)$$

$(\mathtt{E})^n$ stands for the morphological erosion by the unit sphere (or any other suitable structuring element would any suitable a priori information be available), iterated n times.

The number of iterations depends on the size of the input image and on the distribution of the gray levels. The S transform is thus computed on progressively shrunk versions of the binary input image or on steadily intensity reduced versions of the gray level input image, until some predefined decrease or a minimum of intensity is reached.

The iterated elongation, $\eta_n(X)$, is defined as follows:

$$\eta_n(X) = \frac{min_{\alpha \in [0,\pi[}\{IOT_{\alpha,n}(X)\}}{max_{\alpha \in [0,\pi[}\{IOT_{\alpha,n}(X)\}} \qquad (3)$$

It represents dynamic changes of X shapes indicators versus n. Since in most cases, η curves become flat or show some other type of constancy after a certain number of erosions, it was conjectured that any pattern larger than the

structuring element would have a symmetric kernel that IOT reveals: indeed, in eroding a pattern at least one pixel remains to meet the definition. Let us call $IOTK$ this pattern. The intuitive idea here is that the closer the kernel to the pattern, the more symmetric pattern. Unfortunately it is easy to design examples (see Figure 1)where the $IOTK$ is as "far" as one wants from the pattern. Never the less, such a symmetric pattern, bound to the more or less symmetric object under study, should then contribute to a symmetry measure, and more generally to a feeling (sensing) of symmetry by machines.

Remark 1: when it proves necessary, this included version of the kernel could be balanced by the including version obtained by dilation.

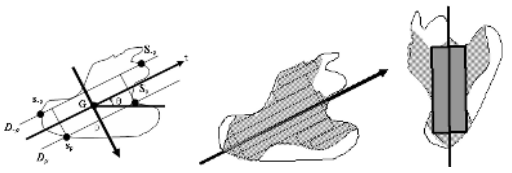
Following commonly used gauges in functional analysis, a possible first answer with a flavor of optimality would be *maximal included symmetric pattern* resp. *minimal including symmetric pattern* : *extremal* then subjects to the measure.

Definition 3. The S-kernel of the pattern P - $SK(P)$ - is the maximal for inclusion symmetric (pattern) subset of P.

A first algorithm to be discussed and optimized is to compute a symmetric pattern included in the given one, *ptrn*, starting from an initial center, G, and iterate the process for all G's and all directions (Figure 1) until the maximum is reached. Here is the main core loop:

For all α
  For all ρ
    Consider
      the symmetric couple $(D_\rho, D_{-\rho})$,
      intersections S_ρ and s_ρ (resp. $S_{-\rho}$ and $s_{-\rho}$) of D_ρ (resp. $D_{-\rho}$)
      with the frontier of *ptrn*
      $S_\rho*$ realizing $\min_{-\rho,\rho} t(S)$
      $s_\rho*$ realizing $\max_{-\rho,\rho} t(s)$
    Let $K_\alpha(ptrn)$ be the union of segments $[s_\rho*, S_\rho*]$ over ρ
    Compute *Symmetry* $(K_\alpha(ptrn))$
Compute $\max_\alpha(Symmetry(K_\alpha(ptrn))$, obtained for $\alpha = \alpha*$
$SK(ptrn) = K_{\alpha*}(ptrn)$

The meaning of S_ρ and s_ρ (resp. $S_{-\rho}$, $s_{-\rho}$) and D_ρ (resp. $D_{-\rho}$) is illustrated in Figure1a. In case of more than 2 intersections (concavities) the algorithm extends by segments in a natural way. Actually the algorithm implementation makes use of the *min* (resp. *max*) operators pixel to pixel in D_ρ and $D_{-\rho}$ respectively, to exhibit the kernel without any prior thresholding (see Figures 3 c and d and 4 b and d) for result samples). Except for some *pathological* patterns and following Definition 1, the center of mass is conjectured a good enough approximation to balance the pattern from. Therefore, not to span the all search space, G is set first to the center of mass of *ptrn* so that the initial state be likely close to the global optimum. The result for the pattern in Figure 1a is the grey



(a) (b) (c)

Fig. 1. (a) Sketch of the kernel detection algorithm; (b) the kernel of the pattern in (a); (c) expanding the $IOTK$ of the pattern in (a) into the kernel

shaded part as given in Figure 1b. One has then to relate the latter kernel to the former indicators ($IOTK$). Let us assume we applied the IOT and found a stable pattern after multiple erosion, like the dark rectangle in the Figure 1c (meaning after that, erosion will confirm symmetry and η remains constant). Starting from there we expand the pattern and mark progressively every where the expansion intersects the border, together with the symmetric pixel wrt. the axis. Every marked pixel is not expanded further. That way the kernel should be obtained again, *provided the center of mass be stable.* The preceding algorithm is a line by line version of the one here, and that makes the expected link.

Remark 2: the center of mass varies from the kernel to the pattern, still all definitions so far assume that tentative symmetry axes pass by this center.

This introduces an additional question: how to define the likely symmetry axis where to compute the kernel from? For instance, let be $\mu = argMaxSymmetry\ (ptrn)$. How does $K_\mu(ptrn)$ compare with $K(ptrn)$? How do their respective Symmetry relate? In most cases $K_\mu(ptrn)$ should be a good enough gauge of $K(ptrn)$, or their difference will be most indicative. The last part of experiments is devoted to answering the question, in checking results over translations of the axis.

In order to test the proposed algorithm we compute a measure of symmetry classically defined as:

$$\lambda = 1 - \frac{Area(D)}{Area(A)}$$

with A, the pattern, B, its kernel, and $Area(D) = Area(A - B)$

It remains a robust first approximation where $\lambda = 1$ if $Area(B) = Area(A)$. In any case, different ways of limiting the pattern should be compared to the one here, that is based on the sole erosion.

4 Experimental Results and Further Comments

In this section we show some results of the S-kernel algorithm applied to synthetic and real images. The purpose of experiments can be summarized as:

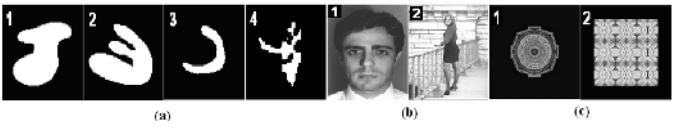
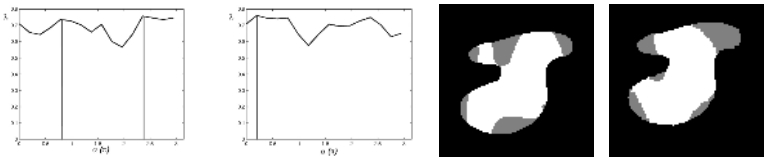


Fig. 2. Sample of images used for experiments: (a) binary; (b) gray level; (c) textured



(a) (b) (c) (d)

Fig. 3. Finding the direction of maximal symmetry for the binary pattern in Figure 2a-1 through G (a) and through shifted centers around G (b) and actual SK superimposed to the input pattern - in white - according to G (c) and the shifted center (d)

1 check the sensitivity of the method to the center position. Two cases will be compared: a) through G; b) shift from G;
2 validate the method ability to measure a degree of symmetry, by comparing $|\eta(SK(ptrn)) - \eta(ptrn)|$ with λ, and $\eta(IOTK)$ with $\eta(SK(ptrn))$;

All experiments are parameterized by direction and completed on both binary and gray level images (see Figure 2).

First, we consider the variation of λ with the angle α used to compute $SK(ptrn)$ on the binary pattern in Figure 2a-1. Figure 3a shows λ versus α when the pole is G; in this example the maximum similarity is $\lambda = 0.76$ and the direction is $\alpha = 135^o$. Figure 3b shows $\lambda_M(\alpha) = max\{\lambda(\alpha, C),\ \ G - S \leq C \leq G + S\}$ when the pole is shifted around G for all α's; the maximum similarity is $\lambda = 0.76$ and the direction is $\alpha = 11.25^o$. The human perception, bound to display, would favor the local minimum near $\pi/2$. The variation of λ for $\alpha = \pi/2$ versus the shift of C shows maximum similarity for a negative five pixels shift from G. White pixels in Figure 3c,d display kernels obtained when $\lambda_{max} = max_\alpha\{\lambda\}$ for fixed and varying centers respectively.We tested the robustness of the algorithms in rotating the images by 45^o; it comes $\lambda_G = 0.76$, $\alpha_G = 45.00^o$ and $\lambda_C = 0.78$, $\alpha_C = 45.00^o$ for image 1b, $\lambda_G = 0.92$, $\alpha_G = 135.00^o$ and $\lambda_C = 0.95$, $\alpha_C = 45.00^o$ for image 2b. Table 1 reports the results for all Figure 2; subscripts G and C indicate respective processing through G and with shifting. The two rightmost columns display the maximum of the Object Symmetry Transform, $OST_{max} = max_{0 \leq \alpha < \pi}\{OST(\alpha)\}$, and corresponding direction, α_{OST}, obtained with the algorithm in [DG3]. Results are comparable, but the kernel algorithm is more accurate with directions. Figures 4a,b show plots of λ_G and λ_C for the image 1c . The ability of the proposed kernel operators is evident in detecting global object symmetries. In fact, the mean values and the variance of λ_G and

Table 1. Comparing kernel results and OST on images in Figure 2

$Image$	λ_G	α_G	λ_C	α_C	OST	α_{OST}
1a	0.76	135.00^o	0.76	11.25^o	0.86	112.50^o
2a	0.74	90.00^o	0.79	33.75^o	0.93	101.00^o
3a	0.82	157.50^o	0.76	22.50^o	0.87	56.25^o
4a	0.76	0.00^o	0.80	0.00^o	0.80	0.00^o
1b	0.80	90.00^o	0.80	90.00^o	0.72	90.00^o
2b	0.70	90.00^o	0.89	90.00^o	0.92	45.00^o
1c	0.99	90.00^o	0.99	135.00^o	0.90	90.00^o
2c	0.99	0.00^o	0.99	90.00^o	0.96	0.00^o

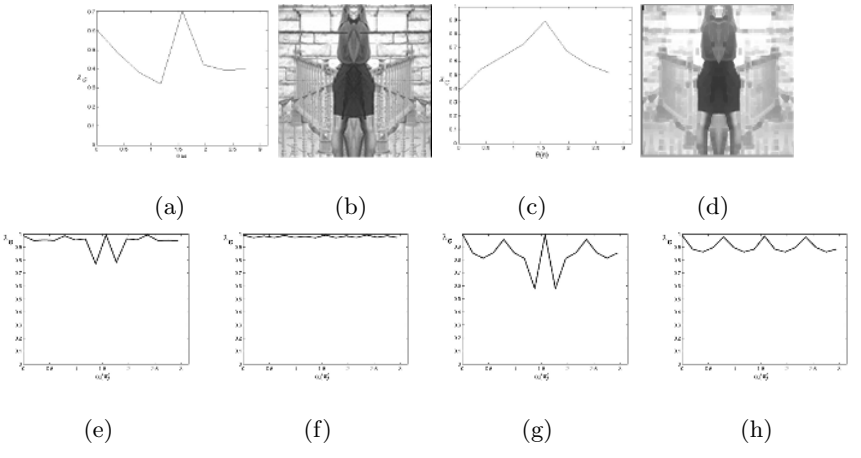


Fig. 4. Plot of λ_G and λ_C for the image 2b (resp. a and c) with corresponding kernels (resp. b and d), and for the image 1c (resp. e and f) and 2c (resp. g and h) of Figure 2

λ_C are $(0.94, 0.2)$ and $(0.94, 0.01)$ respectively indicating the circular symmetry of the image 1c. Note that λ_C is a more robust indicator of the circularity. Same considerations hold for the image 2c (see Figure 4). For comparison, the mean value and variance of the OST of image 1c are $(0.89, 0.01)$ confirming the circular symmetry too.

5 Concluding Remarks

This paper describes a new measure of axial symmetry derived from an object feature named "symmetry-kernel". The symmetry kernel of an object is the maximal subpart that is symmetric regarding a given direction. An algorithm is proposed, based on the computation of the OST for bilateral symmetry. It is rotation invariant and provides quite interesting results. However, it is computationally expensive on conventional computers as it computes line intersections and distances. Eventually, it is sensitive to numerical precision. It was tested on

both synthetic and real data. Experiments show the ability of the symmetry-kernel to detect the main directionality of objects. The distance between an object and its kernel is a crucial point needing further investigation.

References

[CE1] Clarke, F., Ekeland, I.: Nonlinear oscillations and boundary-value problems for Hamiltonian systems. Arch. Rat. Mech. Anal. **78** (1982) 315–333

[BL1] Blum,H., Nagel, R.N.: Shape description using weighted symmetric axis features. *Pattern recognition*, **10**, (1978) 167–180.

[BR1] Brady M., Asada H.: Smoothed Local Symmetries and their implementation. *The Int.J. of Robotics Research*, **3**(3), (1984) 36–61.

[CR1] Cross, A.D.J., Hancock, E.R.: Scale space vector fields for symmetry detection. *Image and Vision Computing*, **17**(5-6), (1999) 337-345.

[DG1] Di Gesù, V., Valenti, C.: Symmetry operators in computer vision. *Vistas in Astronomy*, Pergamon, **40**(4), (1996) 461-468.

[DG2] Di Gesù, V. Valenti, C.: Detection of regions of interest via the Pyramid Discrete Symmetry Transform. *Advances in Computer Vision* (Solina, Kropatsch, Klette and Bajcsy Eds.), Springer-Verlag, (1997).

[DG3] DiGesù, V., Zavidovique, B.: A note on the Iterative Object Symmetry Transform. *Pattern Recognition Letters*, **25**, (2004) 1533–1545.

[KA1] Kazhdan, M., Chazelle, B., Dobkin D., Finkelstein, A., Funkhouser, T.: A reflective symmetry descriptor. 7^{th} *ECCV*, (2002) 642-656.

[KI1] Kiryati, N., Gofman, Y.: Detecting symmetry in grey level images (the global optimization approach). *preprint*, (1997).

[MA1] Marola, G.: On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Trans.of PAMI*, **11**, (1989) 104–108.

[MN1] Manmatha, R., Sawhney, H.: Finding symmetry in Intensity Images", *Tech.Rep.*, (1997).

[OM1] O'Mara, D.: Automated facial metrology, chapter 4: Symmetry detection and measurement. PhD thesis, Feb. (2002).

[OM2] O'Mara, D., Owens, R.: Measuring bilateral symmetry in digital images. *IEEE TENCON*, Digital signal processing applications, (1996).

[SE1] Sewisy, A., Lebert F.: Detection of ellipses by finding lines of symmetry inthe images via an Hough transform applied to staright lines. *Image and Vision Computing*, **19**(12), (2001) 857–866.

[SH1] Shen, D., Ip, H., Teoh, E.K.: An energy of assymmetry for accurate detection of global reflexion axes. *Image Vision and Computing*, **19**, (2001) 283–297.

[FU1] Fukushima, S.: Division-based analysis of symmetry and its applications. *IEEE PAMI*, **19**(2), (1997).

[SH1] Shen, D., Ip, H., Cheung, K.T., Teoh, E.K.:Symmetry detection by Generalized complex moments: a close-form solution. *IEEE PAMI*, **21**(5), (1999).

[BI1] Bigun, J., DuBuf, J.M.H.: N-folded symmetries by complex moments in Gabor space and their application to unsupervized texture segmentation. *IEEE PAMI*, **16**(1), (1994).

[RE1] Reisfeld, D., Wolfson, H., Yeshurun, Y.:Detection of interest points using symmetry. *3rd IEEE ICCV* Osaka, (1990).

[ZA1] Zabrodsky, H.: Symmetry - A review. *Tech.Rep.90-16*, CS Dep. The Hebrew Univ.Jerusalem, 1990.

Easy-to-Use Object Selection by Color Space Projections and Watershed Segmentation

Per Holting and Carolina Wählby

Centre for Image Analysis, Uppsala University, Sweden
carolina@cb.uu.se
http://www.cb.uu.se

Abstract. Digital cameras are gaining in popularity, and not only experts in image analysis, but also the average users, show a growing interest in image processing. Many different kinds of software for image processing offer tools for object selection, or segmentation, but most of them require expertise knowledge, or leave too little freedom in expressing the desired segmentation. This paper presents an easy to use tool for object segmentation in color images. The amount of user interaction is minimized, and no tuning parameters are needed. The method is based on the watershed segmentation algorithm, combined with seeding information given by the user, and color space projections for optimized object edge detection. The presented method can successfully segment objects in most types of color images.

1 Introduction

Object segmentation embrace extracting an object from a non-trivial environment. To be of use for the non-expert, a segmentation tool should be easy to use with as little user interaction as possible. To delineate an object in an arbitrary color image is not trivial. Due to the infinite variation in possible image conditions, no assumptions can be made. The tool has to maximize the use of the information given by the user to try to understand what object is wanted. A Magic Wand is a typical selection tool implemented in many different image-processing softwares, e.g. [1]. Magic Wands are based on pixel similarity in color space, where the tolerated variance is given by the user as a tuning parameter. The tuning parameter is difficult to choose accurately, and repeated interaction is required for segmentation of multi-colored objects, and exact object edges can be difficult to find. The widely spread image processing software Photoshop [1] provides an Extract Tool which works quite well but requires that the user marks the approximate object edge. This may be very time consuming for irregularly shaped objects. Another interactive object selection tool is GrabCut [2]. This tool combines color and edge information, and provides a post-processing step where the edges are smoothed by matting.

1.1 The Presented Approach

The aim of the presented project was to achieve a high performance algorithm at the cost of only modest interaction by the user, yet maximizing the use of

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 269–276, 2005.

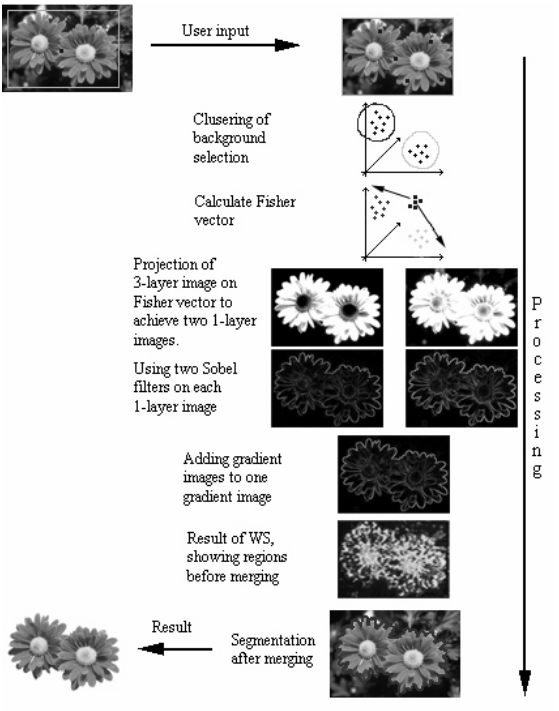


Fig. 1. An overview of the presented algorithm. The gray rectangle and the black dots represent the user input. Each of the steps is described in the method section.

this input. The algorithm is described graphically in Fig. 1. Starting with a color image in the RGB-space the user draws a rectangle around the object to define the background, and a point inside the object to define the object. The rectangle and the point are called the background seed and the object seed respectively. As the background may contain varying colors and textures, the pixels belonging to the background seed are clustered into two clusters. Two projection vectors are thereafter calculated to find the projection that best differentiates between the two background clusters and the object cluster. The gradient magnitudes of the two projected images are summed, and seeded watershed segmentation is applied to the gradient magnitude, using the user input as seeds. The algorithm will find the positions between the background seed and the object seed where the gradient magnitudes are the largest. If the initial result is not satisfying, the user can add more seed points, both background and object seeds, and run the algorithm a second time.

2 Methods

2.1 Image Gradient in Gray-Scale Images

Segmentation algorithms for monochrome images are generally based on one of two basic properties of image intensity values: discontinuity and similarity [3].

Discontinuities in an image are detected by using approximations of first and second derivatives. The first order derivative in a digital context is the gradient where the gradient amplitude is the local edge strength. The derivative of a digital function is often defined in terms of differences. There is a range of ways to define these differences but they all have to fulfill some conditions. The derivative has to be zero in flat, homogenous regions, non-zero at the onset of a grey-level step or ramp and non-zero along ramps. To achieve some smoothing effects and reduce the influence of noise it is appropriate to approximate the gradient operator using a mask consisting of a central pixel and a well defined neighborhood, where each position holds a weight. The Sobel edge detector [3] corresponds to a high pass filter, preserving rapid intensity changes and suppressing small variations and constant regions. A 3 by 3 pixel Sobel edge detector, one for vertical and one for horizontal edges, was used in the presented method.

The gradient calculation discussed above is, unfortunately, not defined for color images, as color images are vector quantities. One approach to this problem is to calculate the gradient for the RGB components separately and add the results [3]. A more sophisticated method is proposed by Di Zenzo [4]. This method calculates the maximum rate of change and lets it influence the final gradient result. We propose a method that optimizes gradient calculation to fit the segmentation seeds given by the user, by combining clustering and Fishers discriminant function.

2.2 Color Clustering

Before reducing the color information, the seeds representing the image background are clustered into two clusters. As the image background may vary on different sides of the same object, clustering before projection and gradient calculation improves the local edge information. More than two clusters would probably improve the result, but two clusters was chosen as a good tradeoff between performance and speed. The k-means algorithm [5] is an exclusive clustering algorithm meaning that data are grouped in exclusive clusters and a certain data point belonging to a defined cluster can only be included in this cluster and no other. The k-means clustering algorithm was used in the presented method, and it follows a simple and easy way to classify a given data set into a given number of clusters, k, fixed a priori. The main idea is to define k centroids, one for each cluster, and associate each data point with a centroid. In the presented approach, the background data is divided into two clusters, and the origin together with the data point at the greates Euclidean distance away from the origin are chosen as initial centroids. Thereafter, k new centroids are found and the data points, again, are associated with the nearest centroids. This loop makes the centroids change their location until no more data points are moved. Doing this, the algorithm aims to minimize an objective function, here the squared error function.

2.3 Fisher's Linear Discriminant

To be able to compute the image gradient with the Sobel filters the 3-layer (RGB-space) image has to be projected onto a 1-layer image. This can be done using

Fishers linear discriminant [6], which will find the projection that maximizes the difference between each of the two background clusters and the foreground cluster, keeping the internal cluster variance small. Data projected on an arbitrary line will often produce a confused mixture of samples from different clusters, but by moving the line around we can find an orientation for which the data is more or less well separated. This way, the dimension can be reduced while the class information is preserved. By calculating the Fisher vector for object seeds and background cluster 1 and for object seeds and background cluster 2, the 3-D problem is converted to a 1-D problem where the difference between the object seed and the two clusters from the background seed is maximized with respect to the projected means and standard deviations. A large difference between object pixel values and background pixel values is an advantage when trying to find edges between the object and the background. It is not trivial to combine the gradients in a way that gives a good segmentation result. The presented approach using Fishers linear discriminant results in a single image where the contrast between object and background is maximized. This means that the gradient separating object from background will also be maximized.

2.4 Watershed Segmentation

A general segmentation algorithm, known as watershed segmentation (WS), was originally presented by Beucher and Lantuèjoul [7]. This method can be applied to different kinds of image information, such as grey-level, distance or gradient information, to divide the image into regions. The method has been refined and used in many situations [8,9]. The WS algorithm can be explained by considering the image as a landscape, where high intensity values in the image correspond to mountains in the landscape, and low values correspond to valleys. Picture drilling a small hole at every local minima, and slowly submerging the landscape into water. The deepest valleys will start to fill with water, and as the water rises, water from different valleys will, in time, meet. At places where water from different valleys is about to meet, a watershed is built to avoid the water to merge. When the whole landscape has been submerged into water, all pixels in the image have been assigned to a region. The WS algorithm is commonly applied to the image gradient magnitude. WS can be implemented by using sorted pixel lists [10] so that essentially only one pass through the image is required. WS often results in many more (or fewer) regions than desired, i.e., over-segmentation or under-segmentation. This can be handled in many different ways, for example by seeding, marking the regions of interest. Seeds can be planted manually or automatically. In the presented method, the user puts seeds in the image background and in the desired object. In addition to the regions represented by the object and background seeds, the initial WS leads to one region per unseeded local minima. Reduction to only two regions is achieved by running a merging algorithm. The merging algorithm merges all non-seeded regions where the difference between the local minima and the corresponding gradient magnitude is the smallest, and the final segmentation will contain object and background regions only.

3 Results

The presented method was tested on hundreds of images, and some of the results are shown below. More results can be viewed at `http://www.cb.uu.se/~carolina/objectselection/`. The results depend on the user input, and thus, the user can influence the result of the segmentation. An unsatisfactory result can also be improved by additional input of seeds by the user. The described method of creating a gradient image by color clustering and projection outperformed other methods for color image gradients when searching for the best gradient image for the desired segmentation. When the object consists of several different areas of color or texture our algorithm needs more than the initial seeds. For relatively simple objects, segmentation can be performed with only one object seed, and no extra user interaction is needed. Fig. 2 shows an exam-

Fig. 2. The girl is the desired object in the image to the left. If a single object seed (grey star on the girl's dress, center image) is used together with a large outer seed, only the girl's dress and hair will be found. By adding a few extra seeds on the girl's legs, hands and chest, the full girl can be extracted (right).

Fig. 3. The edge of the fur of the birds is difficult to find, yet a fairly satisfactory result is achieved by just a single foreground and background seed

ple where a single object seed is not sufficient as the object consists of two very different colors (i.e., the black dress and the pale skin). A satisfactory result is achieved by adding a few more seed points.

Thin smoke, fur and hair in the boundary of objects is not trivial to segment, see Fig. 3. One way of improving the visual result at this type of transparent transitions between object and background is to use a matting technique [2],

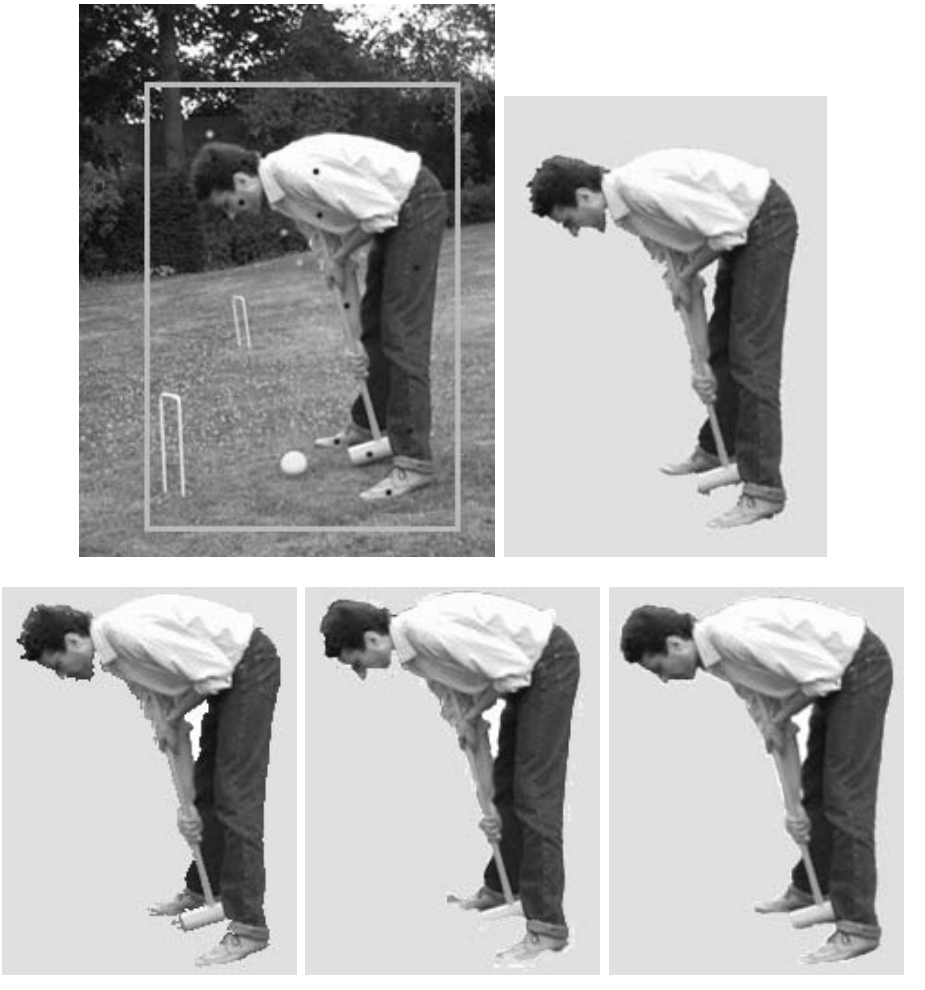


Fig. 4. The croquet player is the desired object in the image in the top left. The result of the presented method, together with input seeds in the top left, is shown to the top right. Below, from left to right, are the results of Magic Wand, Extract Tool, and Grab Cut. Both Magic Wand and Extract Tool need far more user interaction than the presented method. The amount of input needed for GrabCut is not clear, but most likely similar to that of the presented method.

where border-line pixels are given a transparency value based on their similarity to object or background.

Comparing the presented method with other methods shows that our result often is at least as good but with less user input, see Fig. 4. The most difficult areas in this image are at the boundary of the persons head and arm because of a soft transition between object and background, but by adding more seeds a satisfying result is achieved. The Magic Wand requires hundreds of mouse-clicks by the user, even when choosing a proper tolerance. There is also a risk of obtaining small holes within the object due to local color variations. The Extract Tool also manages to segment the object but it requires a careful and time consuming manual tracing of the approximate object boundaries. The presented method is outperformed by the GrabCut algorithm, but this visually more satisfying result is partly achieved by a matting step after the initial segmentation. The result of the GrabCut comes from Microsoft research GrabCut homepage [11], and it is unclear how much user input was required to achieve this result. All test images come from the Berkeley image database [12].

The runtime for the presented method depends much on the size of the processed object, determined by the background seed. For a selection of size 100*250 pixels the algorithm takes about one second and for a larger selection of 200*450 pixels it takes about 10 seconds. The processing time is also dependent on image texture, and can be reduced by smoothing the image before processing, at the price of a lower edge precision. The larger selection (200*450 pixels) with a smoothed image takes about 2 seconds to process. If the computation time is not crucial, the result may be improved by dividing the background seeds into more than two clusters, determined by a clustering algorithm that creates clusters with a specified maximum internal scatter.

4 Conclusions and Future Developments

This paper presents an interactive segmentation tool where foreground and background seeds applied by the user are used for optimized gradient detection and seeded watershed segmentation. The presented method can segment an object in most arbitrary images. In some situations only a rectangle defining background and a single object seed is needed to get a satisfying result. When the object consists of many different regions more user input is needed. For objects with smooth edges, the desired boundary can be found by placing background and object seeds on both sides if the boundary. Highly textured objects are often a problem and much user input is needed, and sometimes no satisfying result can be achieved. For objects with hair at the boundary, the result could probably be improved by post-processing using matting. Taken together, the method is sufficient for object segmentation and does not have any tuning parameters, making it easy to use and appropriate for a general user without expertise knowledge.

Acknowledgements. The authors would like to thank Révolte Development AB who initialized and supported this project.

References

1. Adobe Photoshop version 7.0
2. Rother, C., Kolmogorov, V.,and Blake, A.: GrabCut Interactive Foreground Extraction using Iterated Graph Cuts. ACM Transactions on Graphics **23** 309–314 (2004)
3. Gonzales, R. and Woods, R.: Digital Image Processing, Prentice Hall 2nd edition (2002)
4. Di Zenzo, S.: A Note on the Gradient of a Multi-Image. Computer Vision, Graphics and Image Processing **33**(1) (1986)
5. Tou, J. T. and Gonzalez, R.C.: Pattern Recognition Principles, chapter 3.3.5 Addison-Wesley Publishing Company (1974)
6. Duda, R.O. and Hart P.E.: Pattern Classification and Scene Analysis, chapter 4.10 Wiley-Interscience (1973)
7. Beucher, S. and Lantuèjoul, C.: Use of watershed on contour detection. International Workshop on Image Processing: Real-time and Motion Detection/Estimation, Rennes, France (1979)
8. Meyer, F. Beucher, S.: Morphological segmentation. Journal of Visual Communication and Image Representation, **1**(1) 21–46 (1990)
9. Vincent, L.: Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. IEEE Transactions on Image Processing, **2**,2 176–201 (1993)
10. Vincent, L. and Soille, P.: Watersheds in Digital Spaced: An Efficient Algorithm Based on Immersion Simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence. **13**(6) 583–598 (1991)
11. Microsoft research GrabCut homepage, http://research.microsoft.com/vision/cambridge/segmentation/
12. Berkeley Image database, http://www.cs.berkeley.edu/projects/vision/grouping/segbench/BSDS300-images.tgz

Fast Edge Preserving Picture Recovery by Finite Markov Random Fields

Michele Ceccarelli

Research Centre on Software Technologies-RCOST,
University of Sannio,
Via Traiano 1, 82100 Benevento, Italy

Abstract. We investigate the properties of edge preserving smoothing in the context of Finite Markov Random Fields (FMRF). Our main result follows from the definition of discontinuity adaptive potential for FMRF which imposes to penalize linearly image gradients. This is in agreement with the Total Variation based regularization approach to image recovery and analysis. We also report a fast computational algorithm exploiting the finiteness of the field, it uses integer arithmetic and a gradient descent updating procedure. Numerical results on real images and comparisons with anisotropic diffusion and half-quadratic regularization are reported.

1 Introduction

The Bayesian framework is particularly suited for solving computer vision problems as it can embed in a unique model the data consistency constraints, observation model and a priori assumptions. The underling probabilistic model is the Markov Random Field [5], and it has been successfully applied to several inverse imaging problems such as deconvolution, denoising, interpolation, segmentation, depth estimation, shape from shading and shape from texture. The ill-posed nature of these inverse imaging problems is typically treated by recurring to Gibbs priors encompassing both the uncertainty about the solution and the desirable characteristics it should have. The generic and most popular assumption regards the smoothness of the solution [5,16]. It tends to prefer solutions characterized by local coherence and homogeneity. However, it can lead, in many situations, to over smoothed solution due to the imposition of the constraint everywhere in the image. Indeed, classical image restoration approaches are essentially based on the least squares criteria, which are basically linear and tend to smooth out edges in the output image. Therefore, the application of the smoothness constraint which preserve discontinuities has been one of the most active research areas in the computer vision community [4,11,13,14,15,18]. In particular, the concept of discontinuity adaptive prior (or edge preserving regularization) [11] is becoming even more adopted also due to the availability of fast and accurate algorithms [4,19]. Here we show that the concept of discontinuity adaptive prior can be introduced even in the context of Finite Markov Random Fields (FMRF) where the underlying space of the solution is assumed to be finite. In particular, we classify a potential function as being edge preserving if it treats in the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 277–286, 2005.

same way all the monotone functions in a given interval. This definition avoids to introduce the behavior of the potential function at the infinity and therefore it is more suited for FMRF. We show that in order to be edge preserving, a potential function should weight linearly the image gradient, in agreement with the recent approaches based on the Total Variation norm [3,15]. We also show how to develop a fast computational algorithm for exploiting the finiteness of the field, using integer arithmetic.

2 The MRF Approach

Here we consider the problem of restoring an image corrupted by noise. Let $I^0_{i,j}$, $i = 1, ..., M$ and $j = 1, ..., N$ an observed image and $I_{i,j}$ the "true" image, then our model is

$$I^0_{i,j} = I_{i,j} + n_{i,j} \tag{1}$$

where $n_{i,j}$ denotes the noise. This problem can be solved in the context of Bayesian paradigm. The goal is to estimate the image I^* with the maximum *a posteriori* probability given I_0

$$I^* = \arg\max_I p(I|I_0). \tag{2}$$

It is well known that this MAP estimate can be solved by imposing a constrained problem [6,11,12]:

$$\arg\min_I \mathcal{R}(I) \qquad \text{subject to} \quad ||I - I_0||^2 \leq \sigma^2 \tag{3}$$

where, σ^2 is the noise variance, and $\mathcal{R}(I)$ is the prior energy functional, it measures the "quality of the image" I, in the sense that smaller values of $\mathcal{R}(I)$ correspond to "better" images. $\mathcal{R}(I)$ is the sum of local contribution form each image pixel. When there is no particular knowledge about the kind of images and the specific domain, the most natural assumption about I is its smoothness, therefore, $\mathcal{R}(I)$ should be aimed at measure the irregularities of the solution I, such irregularities being naturally depend on the derivative magnitudes of I.

Classical prior energy functionals are essentially based on the $||\cdot||_2$ norm of the gradient, which has the advantage of producing a set of linear equations to be satisfied by the solution. The main drawback in their use is that these functionals do not allow discontinuities in the solution, i.e. the edges are not well restored. Recently, people is even more interested in edge-preserving methods which produce much better results both from the perceptive point of view and in terms of signal-to-noise ratio. The price to be paid for these advantages is the solution of, sometimes complex, non-linear differential equation arising from the minimum condition of problem (3). In general, the prior energy has the form

$$\mathcal{R}(I) = \sum_{i,j} \phi[(D_x I)_{i,j}] + \sum_{i,j} \phi[(D_y I)_{i,j}]$$

where ϕ is the *potential function*, D_x and D_y are the discretized derivative operators in the x and y directions:

$$(D_x I)_{i,j} = (I_{i,j} - I_{i-1,j})/\delta_x \qquad (D_y I)_{i,j} = (I_{i,j} - I_{i,j-1})/\delta_y.$$

In order to be a suitable potential function, ϕ should satisfy the following general assumptions:

i) $\phi(t) \geq 0$, for any t;
ii) $\phi(t) = \phi(-t)$;
iii) $\phi(t)$ is increasing for $t \geq 0$ and decreasing for $t \leq 0$.

In addition to these assumptions, a potential function ϕ is considered *edge preserving* or *discontinuity adaptive* if it further satisfies [4,11]

iv) $\lim_{t\to\infty} \frac{\phi'(t)}{2t} = 0$;
v) $0 \leq \lim_{t\to 0} \frac{\phi'(t)}{2t} < \infty$.

A number of edge-preserving potential functions have been proposed in literature such as: $\phi(x) = \frac{|x|^\gamma}{2(1+|x|^\gamma)}$ [7]; $\phi(x) = \log(1+t^2)$ [9]; $\phi(x) = \log(\cosh x/\gamma)$ [8] $\phi(x) = |x|$ [15]; $\sqrt{1+t^2} - 1$ [4]; $e^{\frac{-x^2}{\gamma}}$, $\frac{1}{(1+\frac{x^2}{\gamma})^2}$ [14]; and $\phi(x) = \min\{x^2, \gamma\}$ [2].

These conditions are quite natural in the context of images belonging to a continuous framework. However, in practice digital images have values over some finite finite set, such as $\{0, ..., 255\}$. In such case the underlying image model is called *Finite Markov Random Field* (FMRF) representing the fact that $I_{i,j}$ can take only a finite set of values. In this context, the concept of infinity, of course, does not make sense, and condition iv) just represents an ideal behavior. Therefore, successfully edge preserving recovery algorithm should necessarily rely on some additional scale parameter representing thresholds which select candidate edges of the basis of gradient values which are above this threshold. In particular, the study reported in [11] classifies discontinuity adaptive potential functions in terms of the *band*, which is the interval where $\phi''(x) > 0$, outside this interval the penalty term does not depend on x, it can be either zero (no smoothing) or constant as for example the so called *line process potential function* [2]. The above potential functions are typically parametrized by a parameter γ which allow to shrink or expand the *band* of the potential thus allowing a king of smooth threshold for the transition between uniform areas and candidate edges. This parameter being chosen as function of the image scale and the amount of edges one wants to consider inside the image.

Here we want to consider an alternative derivation of the energy potential function which does not depends on the specific values attained by each pixel and therefore is suitable for FMRF.

3 The FMRF Edge Preserving Model

In order to introduce the concept of discontinuity adaptive potential for FMRF let us consider the simple one-dimensional example plotted in Fig. 1 reporting

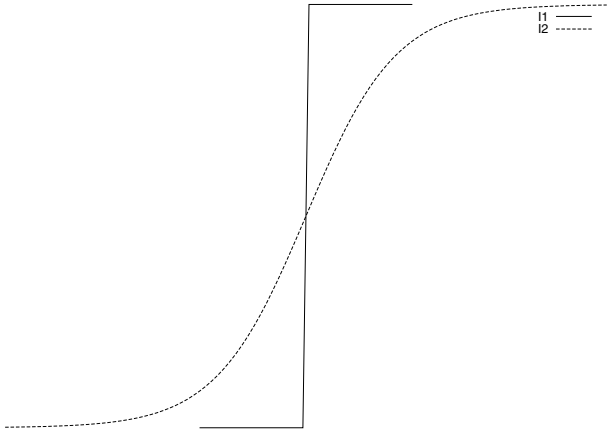


Fig. 1. The function I^1 contains an abrupt change, whereas I^2 is a smooth transition from a value to another

two functions I^1 and I^2, the first containing an evident step discontinuity, the second being a smooth transition from a value to another. Let us consider a discretization of I^1 and I^2 as two sequences $\{I_i^1\}_{i=0,\ldots,N}$ and $\{I_i^2\}_{i=0,\ldots,N}$ with the same discretization step, then

$$\mathcal{R}(I^1) = \sum_{i=1}^{N} \phi(I_i^1 - I_{i-1}^1); \qquad \mathcal{R}(I^2) = \sum_{i=1}^{N} \phi(I_i^2 - I_{i-1}^2).$$

If we want ϕ to be an edge preserving potential then I^1 should not be penalized more than I^2, in the sense that the solution of the problem (3) should not be biased toward I^2, this means that

$$\mathcal{R}(I^1) \leq \mathcal{R}(I^2).$$

This equation guarantees that sharp edges are preserved because the likelihood of solution I^1 is at least as much as that of I^2. In other words, this model does not prefer the smooth behavior of the second solution with respect to the sharp discontinuity of the first. However, if, on the contrary, $\mathcal{R}$ is biased toward I^1 when applied to the image I^2, the solution of (3) will introduce artificial step discontinuities. This could be seen as the lake of the causality principle in the smoothing process aimed at solving problem (3), or equivalently that the smoothing behavior induced by such a potential can introduce artificial features during the regularization process. This last event is particular disastrous in image recovery processes where the aim is to automatic analyze image contents. Therefore we also must have

$$\mathcal{R}(I^1) \geq \mathcal{R}(I^2).$$

From these last two inequalities we derive our definition:

Definition. A potential function ϕ satisfying the general conditions *(i)-(iii)* is said *FRMF edge preserving* if given two monotonically increasing (decreasing) sequences $\{I_i^1\}_{i=0,\ldots,N}$ and $\{I_i^2\}_{i=0,\ldots,N}$ such that $I_0^1 = I_0^2$ and $I_N^1 = I_N^2$ it satisfies

$$\sum_{i=1}^{N} \phi(I_i^1 - I_{i-1}^1) = \sum_{i=1}^{N} \phi(I_i^2 - I_{i-1}^2) \tag{4}$$

Equation (4), according to the above analysis, is the right definition of edge preserving potential for FMRF. Actually it does not make use of its behavior at infinity, rather, it is based on the weight it gives to similar sequences which eventually contain abrupt changes representing edges. Neither it requires the choice, in terms of appropriate thresholds, of what a discontinuity is. It is our aim, now, to characterize the properties a function should satisfy in order to be a FMRF edge preserving potential. The first consequence of our definition is that among the potential functions listed in table 1 the Total variation norm [15] is FMRF edge preserving.

Theorem 1. *A FMRF edge preserving potential $\phi(x)$ is a linear function of x for $x \geq 0$.*

Proof. Let us consider an increasing sequence $\{f_i^1\}_{i=0,\ldots,N}$, $N > 1$, and two integers ξ and η such that $0 < \xi < \eta \leq N$. Set

$$f_i^1 = \begin{cases} \mathrm{b} & i \geq \eta \\ \mathrm{a} & \xi \leq i < \eta \\ 0 & i < \xi \end{cases}$$

where $a, b \in \mathbf{R}$, with $a < b$. Now, let us define another increasing sequence f_2 as follows

$$f_i^2 = \begin{cases} \mathrm{b} & i \geq \eta \\ 0 & 0 \leq i < \eta \end{cases} .$$

Both f^1 and f^2 are increasing, therefore $f_i^1 - f_{i-1}^1 \geq 0$ and $f_i^2 - f_{i-1}^2 \geq 0$, moreover $f_0^1 = f_0^2$ and $f_N^1 = f_N^2$, then let $c = b - a$, if ϕ is a FMRF edge preserving than

$$\phi(a + c) = \phi(b) = \sum_i \phi(f_i^2 - f_{i-1}^2) = \sum_i \phi(f_i^1 - f_{i-1}^1) = \phi(a) + \phi(b - a) = \phi(a) + \phi(c)$$

and this is true for any $a \geq 0$ and $c \geq 0$.

This result is, of course, not surprising. For example, most of the edge preserving functions propsed in literature have a linear behavior at infinity such as [8] [15] and [4]. In addition to the edge preserving property these three functions have the nice property of convexity, which is of help in the solution of (3). Our derivation, however, sates that in order to have an edge preserving potential, this linear behavior should be always satisfied, clarifying what implicitly stated by condition *(iv)*.

3.1 Computation of a FMRF Edge Preserving Solution

Given an FMRF edge preserving potential we want to show how we can solve problem (3) with a fast and efficient algorithm. Let us call $\mathcal{G} = \{g_0, ..., g_{L-1}\}$ the finite set where the image pixels take values, *i.e.* $I_{i,j} \in \mathcal{G}$ and consider the maximum difference between two image values:

$$\Delta = \min_{k \neq l} |g_k - g_l|.$$

Here we develop a simple iterativa algorithm aimed at the minimization of the discrete functional $\mathcal{R}$ with the given constraint by following the iterative scheme

$$\begin{aligned} I_{i,j}^{n+1} = I_{i,j}^n + \Delta \cdot \text{sign}[&\text{sign}(I_{i+1,j}^n - I_{i,j}^n) - \text{sign}(I_{i,j}^n - I_{i-1,j}^n) \\ &+ \text{sign}(I_{i,j+1}^n - I_{i,j}^n) - \text{sign}(I_{i,j}^n - I_{i,j-1}^n)] \end{aligned} \tag{5}$$

this scheme is iterated while $||I^{n+1} - I^0|| \leq \sigma$ is true. Since $\mathcal{R}(I) \geq 0$, the following proposition states the convergence of the scheme.

Theorem 2. *The sequence of potentials $\mathcal{R}(I^n)$ generated by scheme (6) decreases monotonically.*

The proof easily follows by considering all the possible configurations (which are finite) in the neighborhood of each pixel and will be reported elsewhere. The next proposition states the causality property, which is fundamental for every iterative smoothing process. Roughly speaking, the causality principle states that each feature at a coarse scale must have a cause at a finer scale. This means that the smoothing process does not introduce spurious features. Formally, it can be shown that every causal smoothing process must be governed by, or be the discretized version of, a parabolic partial differential equation obeying a maximum principle [1].

Theorem 3. *The scheme (6) satisfies*

$$\min\{I_{i,j}^n, I_{i-1,j}^n, I_{i+1,j}^n, I_{i,j-1}^n, I_{i,j+1}^n\} \leq I_{i,j}^{n+1} \leq \max\{I_{i,j}^n, I_{i-1,j}^n, I_{i+1,j}^n, I_{i,j-1}^n, I_{i,j+1}^n\}$$

Proof. Let $m = \min\{I_{i,j}^n, I_{i-1,j}^n, I_{i+1,j}^n, I_{i,j-1}^n, I_{i,j+1}^n\}$, and $M = \max\{I_{i,j}^n, I_{i-1,j}^n, I_{i+1,j}^n, I_{i,j-1}^n, I_{i,j+1}^n\}$ if $I_{i,j}^n = m$ then it is easy to verify that

$$[\text{sign}(I_{i+1,j}^n - I_{i,j}^n) - \text{sign}(I_{i,j}^n - I_{i-1,j}^n) + \text{sign}(I_{i,j+1}^n - I_{i,j}^n) - \text{sign}(I_{i,j}^n - I_{i,j-1}^n)] \leq 4$$

and, in this case, from (6)

$$I_{i,j}^{n+1} = I_{i,j}^n + \Delta \geq m.$$

In the other cases, from the definition of Δ, we have

$$I_{i,j}^n \geq m + \Delta$$

and therefore

$$I_{i,j}^{n+1} \geq I_{i,j}^n - \Delta \geq m$$

where the first inequality comes from (6). The proof that $I_{i,j}^{n+1} \leq M$ is analogous.

4 Experiments and Comparisons

In this section we will present some experimental result about the application of the developed algorithm to synthetic and real grayscale. In particular the algorithm reads as:

Digital Picture Recovery Algorithm

//Input: a discrete image $I^0_{i,j}, i = 1, ..., M$ and $j = 1, ..., N$
//Output: the recovered image

1. Estimate $\tilde{\sigma}$
2. $I_{i,j} := I^0_{i,j}$
3. $n := 0$;
4. **while** $(\sum_{i,j}(I^n_{i,j} - I^0_{i,j})^2 \leq MN \cdot \sigma)$
5. Apply (6) to each image pixel
6. $n := n + 1$
7. **end while**
8. output I^n.

Note that there are several methods to perform step 1. Our implementation adopts a variant of the method proposed in [10]. In particular,

$$\tilde{\sigma} = \frac{1}{36}\text{Variance}(I^0 \otimes \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix}) \tag{6}$$

where $\otimes$ represents the convolution operator. Note that the algorithm does not require any parameter. In our implementation we have the choice to implement filter (6) in a recursive manner, *i.e.* the updating is performed in place, this kind of updating produces a significant speed-up of the convergence while maintaining the causality properties of the method. In any case the experiments presented below are based on batch updating. Since we also report computing time, the adopted computing platform is significant, all the experiments were performed on a 600 MHz Pentium II Linux Personal Computer.

In order to evaluate the behavior of the algorithm, and to compare it with other edge preserving denoising, we artificially add to the original image some amount of noise and then measure the quality of the reconstruction as function of the iteration. Here we compare the algorithms with well known edge preserving image recovery techniques such as the half-quadratic regularization by the ARTUR algorithm [4,6] and anisotropic diffusion [3,14,15]. It is well known that the quantitative measures of image reconstruction may often fail with respect to perceptually plausible measures. For example the mean squared error measure tends to compress small errors and to overweight large errors. In this paper we adopt as a quantitative measure of the reconstruction the so called *Mean Error* (ME) defined as

$$ME(I, I^0) = \frac{1}{n}\sum_{i,j} |I_{i,j} - I^0_{i,j}| \tag{7}$$

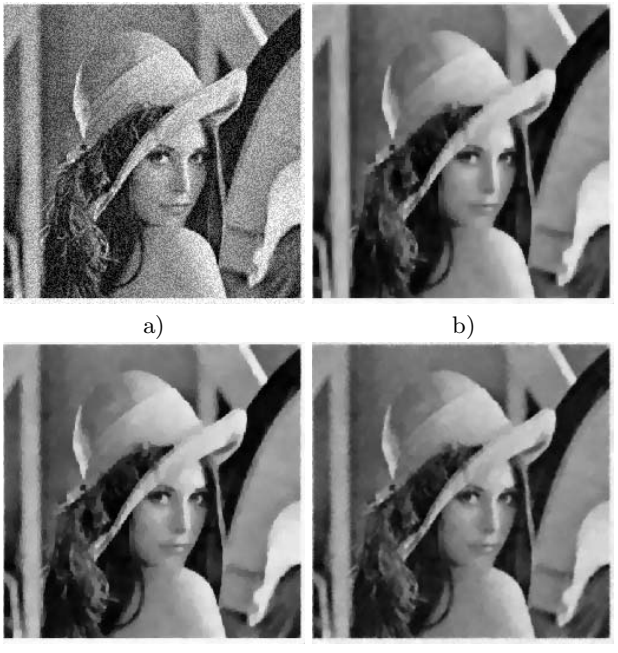


Fig. 2. The lena image, corrupted by uniform noise at 8.5 dB of SNR a), and its recostruction by the proposed method b), the anisotropic diffusion c) and half quadratic regularization d) ($\alpha = 0.075$)

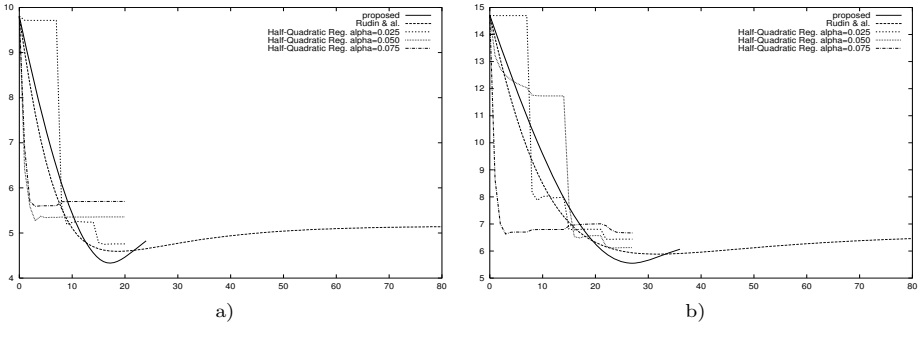


Fig. 3. The ME measure as funtion of the iteration for the reported algorithms. For the half quadratic regularization the measure is computed at each iteration of the iterative algorithm adopted to solve the inner linear system, which our case is a conjugate gradient algorithm with a multigrid preconditioner. This figure refers to uniform noise at 14.5 dB a) and at 11 db b) of SNR.

where n is the number of image pixels. The first case adopts uniformly distributed additive noise. In particular we add uniform noise at an amount of 8.5dB, 11dB and 14.5dB of Signal to Noise Ratio (SNR). The corrupted image and the corresponding reconstruction are reported in Fig. 2. As it can be seen from the images there is no significant difference between the reconstructions, at least from the perceptual level. In order to quantitatively appreciate the behavior of the algorithm we report the ME as function of the iteration number. As Fig. 3 shows, the proposed algorithm compares well in terms of quality of reconstruction with the other algorithms reported. For what concerns the parameters adopted for the generation of such figure let us mention that our algorithm does not need any free parameters, whereas for the case of anisotropic diffusion we choose the time step as 0.5, the maximum number of iteration is 80 and the noise variance the same that estimated by (6). Whereas, for the half-quadratic reconstruction we adopted the regularization parameters reported in the figure, and fixed the maximum number of outer iterations to four and the maximum number of inner iteration of the preconditioned conjugate gradient algorithm to seven. The above figures also show that the ARTUR algorithm has a very fast convergence rate however one should consider that the price in terms of computation is much higher with respect to the proposed algorithm. Specifically, for the reported 256×256 grayscale image the computing times over the adopted platform of each inner iteration, are 0.027, 0.076 and 0.562 seconds respectively for the proposed algorithm, an isotropic diffusion and half-quadratic regularization. This means that each inner iteration takes 5% of the the time of half quadratic algorithm, and 14% of the time of each non linear diffution iteration. If, in addition, we consider that the right regularization parameter must be typically chosen in an experimental trial and error fashion, the advantage of the proposed method is even more evident.

5 Conclusions

We have reported an image recovery algorithm which is based on the Finite Markov Random Field model. We have investigated the properties of edge preserving potential functions for FMRF and clarified that the linear behavior of potential functions is fundamental for convex edge preserving priors. The resulting algorithm is fast and efficient, does not require any choice of free parameters.

References

1. L. Alvarez, P. L. Lions, F. Guichard, and J. M. Morel, "Axioms and Fundamental equations of Image Processing", *Archives for Rational Mechanics and Analysis*, vol. 16(9), pp. 200-257, 1993
2. A. Blake, and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, MASS., 1987.
3. M. Ceccarelli, V. De Simone, "Well Posed Anisotropic Diffusion for Image Denoising", em IEE Proceedings Proceedings-Vision, Image and Signal Processing, vol 149, 4, pp. 244-252, 2002.

4. P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic Edge-Preserving Regularization in Computed Imaging", *IEEE Transactions on Image Processing*, vol. 5, pp. 298-311, 1997.
5. S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution and the Bayesian Restoration of Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721-741, 1984.
6. S. Geman and G. Reynolds, "Constrained Restoration and the Recovery of Discontinuities", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 367-383, 1992.
7. S. Geman and D. E. McClure, "Bayesian image analysis: an application to single photon emission tomography", in *Proc. Statistical Computational Section, Amer. Statistical Assoc.*, Washington, DC, 1995, pp. 12-18.
8. P. J. Green, "Bayesian reconstruction for emission tomography using a modified EM algorithm", *IEEE Transactions on Medical Imaging*, vol. 9, pp. 84-93, 1990.
9. T. Hebert and R. Leahy, "A Generalized EM Algorithm for 3-D Bayesian Reconstruction form Poisson Data using Gibbs Priors", *IEEE Transactions on Medical Imaging*, vol. 8, pp. 194-202, 1990.
10. J. Immerkaer, "Fast Noise Variance Estimation", *CVGIP: Image Understanding*, vol. 64, pp. 300-302, 1996.
11. S. Z. Li, "On Discontinuity-Adaptive Smoothness Priors in Computer Vision", *IEEE Transactions on Pattern Analysis and machine Intelligence*, vol. 17, pp. 576-586, 1995.
12. S. Z. Li, *Markov Random Field Modeling in Computer Vision*, Springer, Berlin, 1995.
13. D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems", *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577-685, 1989.
14. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffution', *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 345-362, 1990.
15. L. Rudin, S. Osher and E. Fatemi "Nonlinear total variation noise removal algorithms", *Physica D*, vol. 60, pp. 259-268, 1992.
16. T. Poggio, V. Torre, C. Koch, "Computational Vision and Regularization Theory", *Nature*, vol. 317, pp. 314-319, 1985.
17. Y. L. You, W. Xu, A. Tannebaum and M. Kaveh, 'Behavioral Analysis of Anisotropic Diffution in Image Processing', *IEEE Transactions of Image Processing*, vol. 5, pp. 1539-1553, 1996.
18. D. T. Terzopoulos, "Regularization of inverse visual problems involving discontinuities", *IEEE Transactions on PAMI*, vol. 8(4), pp. 413-242, 1986I.
19. C. R. Vogel and E. Oman, 'Iterative Methods for Total Variation Denoising', *SIAM Journ. on Scientific Computing*, vol 17, pp. 227-238, 1996.

High Speed Computation of the Optical Flow

Hiroaki Niitsuma and Tsutomu Maruyama

Systems and Information Engineering, University of Tsukuba,
1-1-1 Ten-ou-dai Tsukuba Ibarakim, 305-8573, Japan
niitsuma@darwin.esys.tsukuba.ac.jp

Abstract. In this paper, we describe a compact system for high speed computation of the optical flow. This system consists of one off-the-shelf PCI board with one Field Programmable Gate Array (FPGA) chip, and its host computer. With this system, we can generate dense vector maps at (1) 840 frames per second (fps) in small size (320×240) images, and (2) 30 fps in standard size (640×480) images by configuring different circuits on the FPGA chip. In the two circuits, vectors for all pixels in the images are obtained by the area-based matching (windows of 7×7 pixels are compared with 121 and 441 windows in the target image respectively). The circuits implemented on the FPGA do not require any special hardware resources, and can be implemented on many off-the-shelf FPGA boards shipped from many vendors. This system can also be used for the stereo vision by slightly modifying the circuits, and achieve the same performance.

1 Introduction

Compact vision systems are very important for autonomous vehicles. Field Programmable Gate Arrays (FPGAs) are ideal devices for the compact systems, because any kinds of circuits can be realized on FPGAs by just downloading configuration data to FPGAs from external memories or host computers (loading time is 10 to 100 msec in general). Depending on situations, autonomous vehicles may try to reconstruct the 3-D geometry to understand its circumstances, to find out moving objects to move safely, and to find out marker objects to check its position. FPGAs can support all these functions by reconfiguration.

In this paper, we describe a compact system for the optical flow which consists of an off-the-shelf PCI board with one FPGA chip and its host computer (to download configuration data and display the results). In our system, the most similar parts to small windows ($w \times w$ pixels) in one image are looked up in the next image taken by the same camera to obtain the optical flow. In these comparison of small windows, the SAD (Sum of Absolute Difference) algorithm is used because of its simplicity. The amount of the computation in the optical flow is, however, still very large because of the two dimensional search to find out objects moving to all directions, and high speed computation is not easy even on the latest LSIs owing to the limited memory bandwidth.

We implemented two kinds of circuits on the FPGA chip. In the first implementation, intermediate results in the computation along x and y axes are

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 287–295, 2005.

stored on the chip and reused w times (but part of them are recalculated in order to minimize the amount of data which have to be stored on the chip) in order to achieve highest performance on small size (320×240) images. In the second implementation, intermediate results along x axis are stored, but operations along y axis are re-executed w times in order to minimize the circuit size while maintaining video-rate processing on standard size images (640×480).

This system can also be used for the stereo vision by slightly modifying the circuits, and it becomes possible to detect moving objects in images taken by moving cameras by combining the stereo vision with the optical flow.

2 The Optical Flow

In an image taken by a camera, each pixel corresponds to the intensity value obtained by the projection of an object in 3-D space onto the image plane. When the object or the camera moves, its corresponding projection also changes position in the image plane. Optical flow is a vector field that shows the direction and magnitude of these intensity changes from one image to the other. In the optical flow, the corresponding point to a given point in an image is searched in the next image taken by the same camera. Area-based (or correlation-based) algorithms match small windows centered at a given pixel to find corresponding points between the two images. They yield dense maps, but fail within occluded areas (occlusions are caused by the movement of the camera). Feature-based algorithms match local cues (e.g., edges, lines, corners) and can provide robust, but sparse maps which require interpolation. In hardware systems, area-based algorithms are widely used, because the operations required in those algorithms are very regular and simple.

The most common pixel-based matching algorithm is squared intensity differences (SSD) and absolute intensity differences (SAD). We used the SAD (Sum of Absolute Difference) algorithm because it is the simplest, and its result is almost same as other algorithms in the stereo vision[1]. In the SAD algorithm for the optical flow, ξ and η which minimize the following equation are searched.

$$SAD(x, y, \xi, \eta) = \sum_{i=-w/2}^{w/2} \sum_{j=-w/2}^{w/2} |I_0(x+i, y+j) - I_1(x+i+\xi, y+j+\eta)|$$

In this equation, I_0 and I_1 are images in $time = t$ and $time = t + \Delta t$ respectively, and $w \times w$ is the size of the window centered at a given pixel (its position is (x, y)). The range of ξ and η decides the size of area where the corresponding point to (x, y) is searched.

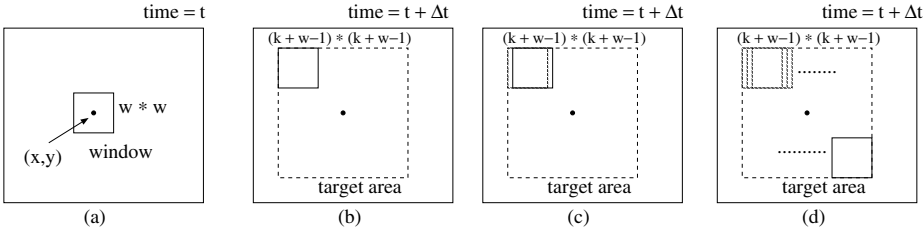


Fig. 1. Area-Based Matching in the Optical Flow

In Figure 1, a small window centered at (x, y) (Figure 1(a)) is compared with all windows in its target area centered at (x, y) (Figure 1(b)(c)(d)). When the size of the target area is $(k + w - 1) \times (k + w - 1)$, there are $k \times k$ windows in the target area, and $k \times k$ SADs (Sum of Absolute Differences) are calculated. Then, the window which gives the minimum SAD is chosen, and its center point (x', y') is considered as the corresponding point to (x, y). In this comparison, every pixel in the window in $time = t$ is compared with $k \times k$ pixels in the target area (the range of ξ and η is $-k/2$ *to* $k/2$). By this two-dimensional search, we can obtain one vector from (x, y) to (x', y').

3 Previous Works

Many approaches to reduce the computational complexity of the optical flow have been proposed[2][3], but in those algorithms, computations of areas which seem to be unnecessary for detecting moving objects are not executed, and users need to think of trade-offs between accuracy and efficiency.

In order to accelerate its performance by hardware, many systems have been proposed to date[4][5][6][7]. In those systems, in order to achieve real-time processing, sizes of images are limited or only sparse vector fields are generated. Their performances are, however, still slower than video-rate in the standard size images.

4 Computation Methods of the Optical Flow on the FPGA

Suppose that the size of the images is $X \times Y$, and N pair of images are processed in one second. Then, in order to find vectors for all pixels in the images, we have to calculate $X \times Y \times N$ vectors in one second. This means that only 108 nano seconds is allowed to find out one vector, when the image size is 640×480 and N is 30. Furthermore, $k \times k$ SADs have to be calculated for one vector, and $w \times w$ ADs (Absolute Differences) are necessary for calculating each SAD. This requirement means that we need to reuse intermediate results generated during the calculation of a window for the calculations of other windows. In the following, we first show a technique to realize the maximum performance, then we discuss another technique to compare a window with more number of windows in the target area at the video-rate.

4.1 A Technique to Realize the Maximum Performance

In order to achieve maximum performance on hardware, all operations have to be processed in parallel and in pipeline. Therefore, suppose that all operations described in this subsection are executed in parallel and in pipeline.

In Figure 2(a), suppose that we have calculated $k \times k$ SADs ($k \times k \times w \times w$ ADs (Absolute Differences) have been calculated) and chosen the minimum of them to obtain one vector (computations of only two SADs are shown to simplify the figure). During these computations, no operations on same data are executed.

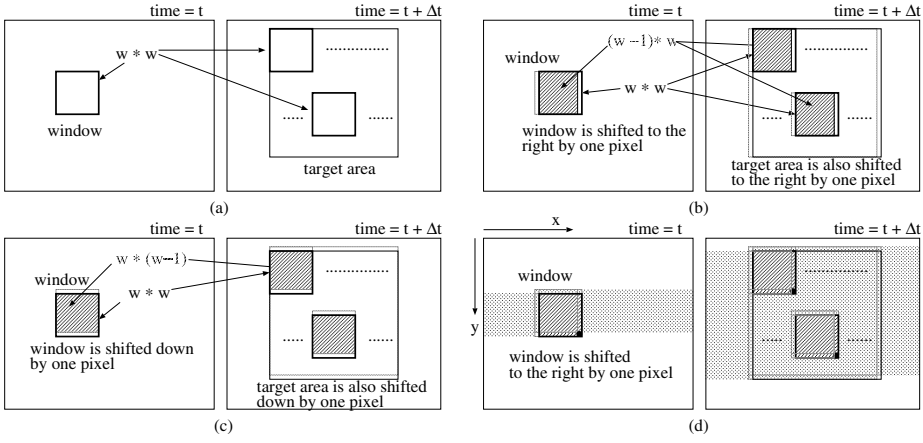


Fig. 2. Reuse of the Intermediate Results

Then, the window is shifted to the right by one pixel to obtain next vector (Figure 2(b)). At this point of time, pixels in a rectangle with slanting lines in the shifted window ($time = t$) are already compared with pixels in rectangles with slanting lines in its target area ($time = t + \Delta t$) during the computation of the previous vector. Therefore, by storing $k \times k \times (w-1) \times w$ ADs (Absolute Differences) calculated in Figure 2(a), the number of new ADs to obtain the new vector can be reduced to $k \times k \times w$. When the window is shifted down by one pixel as shown in Figure 2(c), pixels in a rectangle with slanting lines in the shifted window are already compared with pixels in rectangles with slanting lines in its target area. In this case, we can also reduce the number of AD operations to $k \times k \times w$ by storing and reusing the $k \times k \times w \times (w-1)$ ADs.

In Figure 2(d), suppose that the image size is $X \times Y$ and the window is shifted to the right (along x axis) first, and when the window reaches to the right-end of the image, the window is moved to the left-end again and shifted down by one pixel. In this case, when the window is shifted to the right by one pixel, ADs for $w \times w - 1$ pixels in the shifted window (all pixels in the window except for one pixel shown by a black dot) are already calculated ($k \times k \times (w \times w - 1)$ ADs are already calculated) during the computation of previous vectors. Therefore, by storing $k \times k \times (w-1) \times X$ ADs (which correspond to the gray area in Figure 2(d)), we can calculate $k \times k$ SADs which are necessary to obtain a new vector by only calculating $k \times k$ new ADs (ADs between the pixel shown by the black dot and $k \times k$ pixels in the target area). In this computation, we need to access $k \times k \times (w \times w - 1)$ ADs (which are already calculated and stored) in parallel in order to achieve maximum performance.

Figure 3 shows an implementation technique to make the parallel access possible (the upper half of the figure shows the two images which are compared, and the lower half of the figure shows an array of SAD units, and the inside of a SAD unit). In Figure 3, suppose that the vector for a pixel (light gray square in the window) was just obtained, and the window is shifted to the right to find the

vector for the next pixel (dark gray square). Then, the window is compared with $k \times k$ windows in its target area ($k \times k$ SADs are calculated), and the minimum SAD is searched. In order to achieve maximum performance, $k \times k$ SAD units are prepared and the $k \times k$ SADs are calculated in parallel (In Figure 3, only two units are shown to simplify the figure). In Figure 3, $A_{i,j}$ are ADs (Absolute Differences) which are already calculated during the computation of previous vectors. In Figure 3, a new SAD is calculated using $A_{i,j}$ as follows.

1. w-1 ADs ($A_{i,6}(i = 2, 5)$ (squares with sparse slanting lines)) are read out from memory M_A.
2. A new AD for the black square (which becomes $A_{6,6}$) is calculated (pixel data of the black square (I_4) is broadcasted to all SAD Units on the array).
3. These w ADs are held on w shift registers in the SAD unit. Each shift register can hold w ADs (w is 5 in Figure 3). Thus, $w \times w$ ADs are on the shift registers in total. The ADs on the shift registers are shifted when a new SAD (consequently a new vector) is obtained.
4. These $w \times w$ ADs on the shift registers are summed up to calculate a new SAD.
5. Among w ADs which are shifted out from the shift registers, w-1 ADs are written back to M_A ($A_{i,1}(i = 3, 6)$ (squares with dense slanting lines)). Thus, each AD is summed up w times while it is on the shift register, and is stored and read out from the memory w-1 times, which means each AD is used for calculating $w \times w$ SADs.

By repeating the procedure above with $k \times k$ SAD units which run in parallel and in pipeline, we can continue to obtain a new vector in every clock cycle.

In this implementation, the width of memory M_A must be w-1 words. Therefore, the total number of memory banks required in this implementation becomes $k \times k \times (w - 1)$, and these memory banks must be accessed in parallel. This means that these memory banks have to be located on the FPGA (because the input/output performance of LSIs (including FPGAs) is very limited). However, the number and width of internal memory banks of the latest LSIs are not enough under the practical w and k.

In order to reduce the number of memory banks, the procedure described above is modified as follows.

1. Only the sum of the w-1 ADs is stored in the memory (suppose that $\sum_{i=2}^{5} A_{i,6}$ is in the memory).
2. The sum is read out, and added with $A_{6,6}$ ($\sum_{i=2}^{6} A_{i,6}$ can be obtained).
3. w sums on a shift register are summed up to calculate new SAD ($\sum_{i=2}^{6} \sum_{j=2}^{6} A_{i,j}$).
4. At the same time, $A_{2,6}$ is calculated again, and subtracted from $\sum_{i=2}^{6} A_{i,6}$.
5. The result ($\sum_{i=3}^{6} A_{i,6}$) is stored in the memory to obtain vectors on the next row.

With this technique, we can reduce the number of memory banks to $k \times k$ from $k \times k \times (w - 1)$, though we need double SAD units. The total hardware resources required by this technique are $k \times k \times 2$ SAD units, $k \times k$ memory banks and a unit to choose the minimum among $k \times k$ SADs in pipeline.

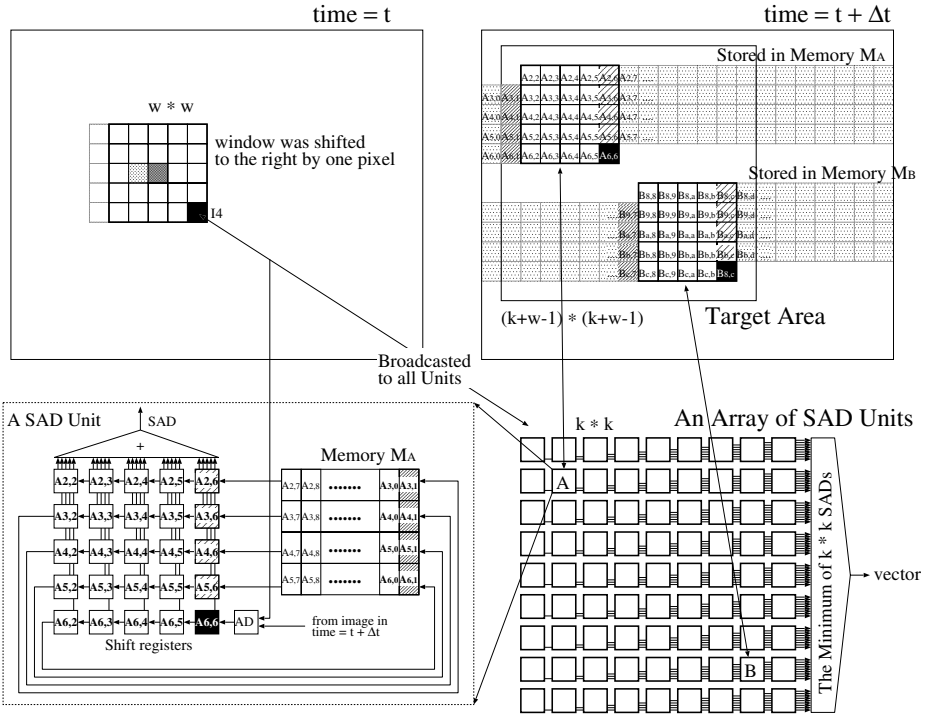


Fig. 3. An Implementation Technique to Achieve Maximum Performance

4.2 Video-Rate Processing

We can reduce the circuit size by recalculating some of ADs instead of storing all of them. This approach makes it possible to enlarge the size of the target area.

By recalculating sums of ADs which were given by the memory banks ($\sum_{i=2}^{5} A_{i,6}$ in Figure 3), we can calculate SADs without memory banks. In Figure 4, I_0 is broadcasted to $k \times k$ SAD units first, and $k \times k$ ADs for I_0 ($|A_{2,6} - I_0|$ in Figure 3) are calculated in the $k \times k$ SAD units in parallel. In the same way, ADs for $I_j (j = 1, 4)$ are calculated sequentially. These calculations take w clock cycles in total. These ADs are, then, summed up, and held on the shift registers. The sums held on the shift registers are used w times to calculate w SADs and discarded after shifted w times.

Though this implementation requires w clock cycles to generate one vector, we can calculate vectors with $k \times k$ SAD units, no memory banks and a unit to choose the minimum among $k \times k$ SADs in pipeline. Furthermore, the size of the unit to choose the minimum SAD can be reduced to almost $1/w$, because many parts of the unit can be shared by w SAD units (each SAD unit generates one SAD in every w clock cycles).

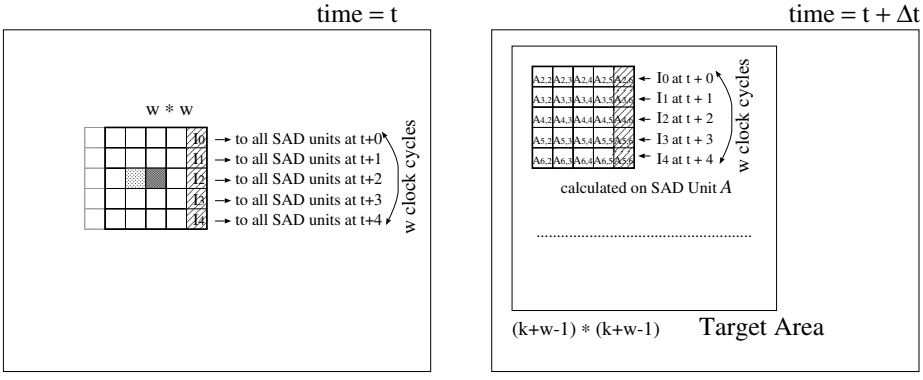


Fig. 4. An Implementation Method by Recalculation

The requirement for the video-rate processing is to obtain one vector in 108 nano seconds. Therefore, if we can build a circuit which runs faster than $108/w$ nano seconds, we can realize video-rate processing by this implementation method. The typical w used for the area-based matching is 7. Therefore, our goal is to build a circuit which runs faster than 65MHz.

5 Performance

Two kinds of circuits were implemented on PCI board (ADM-XRC-II by Alpha Data [9]) with one FPGA (Xilinx XC2V6000 [10]). Both circuits run at 66 MHz. Table 1 shows the hardware usage and the performance of the two circuits. The maximum performance of our camera is 30 frames per second. Therefore, we could not demonstrate higher frame rates than that on our system, but we confirmed that our circuits can process a pair of images at the speeds which are shown in Table 1.

The value of k in the circuit-1 (the circuit for the maximum performance) is much smaller than the circuit-2 (the video-rate circuit), because $k \times k \times 2$ SAD units are required in the circuit-1, and any part of the unit for choosing the minimum SAD can not be shared. 7 memory banks in the circuit-1 and 16 memory banks in the circuit-2 are used as cache memory to store a part of images in order to reduce data access to external memory banks on the FPGA board.

The computation time of the both circuits is promotional to the image size. The performance of the circuit-2 (the video-rate circuit) decreases proportional to w because same operations are executed w times in this implementation technique, while the performance of the circuit-1 (the circuit for the maximum performance) is constant. The size of the both circuits is almost proportional to $k \times k$, and w.

Figure 5 shows an example of the output by the circuit-2 (only a part of vectors are shown in the figure, because the image size is 640×480 and dense

Table 1. Performance of the Circuits

	k	usage of hardware resources		frame per second	
		memory banks	logic blocks	640 × 480	320 × 240
circuit-1 (max. performance)	11	128	71%	210	840
circuit-2 (video-rate)	21	16	84%	30	120

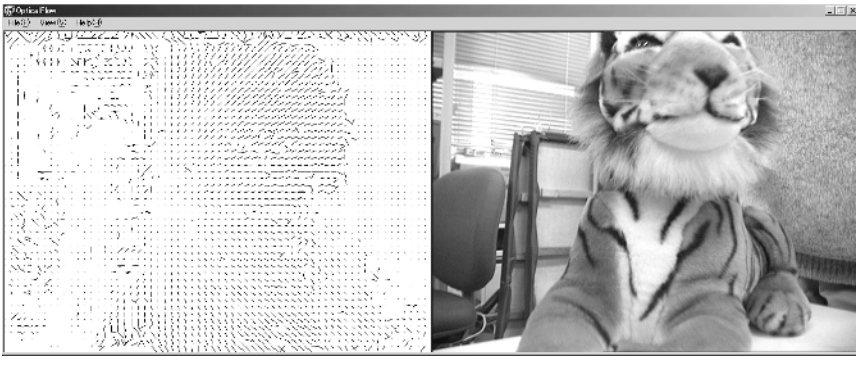


Fig. 5. The Output by the Circuit

vector map is generated by the circuit). In Figure 5, a stuffed toy (tiger) is moving to the right, but some noises are found on areas with only small changes in the contexts. We need to add some circuits to suppress these noises.

6 Conclusions

In this paper, we described a compact system for high speed computation of the optical flow. The system was implemented on an off-the-shelf PCI board with one FPGA. With this system, we could generate dense vector maps at (1) 840 frames per second (fps) in small size (320 × 240) images, and (2) 30 fps in standard size (640 × 480) images by configuring different circuits on the FPGA chip. In the two circuits, vectors for all pixels in the images are obtained by the area-based matching (windows of 7 × 7 pixels are compared with 121 and 441 windows in the target image respectively).

We are now improving the system to work with (1) the stereo vision to detect moving objects in the images taken by moving cameras, and (2) edge detections to clearly distinguish borders of the moving objects.

References

1. T.Kanade, "Development of a video-rate stereo machine", IUW, pp. 549-557. 1994.
2. Camus, T.A., "Real-Time Quantized Optical Flow", Workshop on Computer Architectures for Machine Perception 1995
3. Zelek, J.S., "Bayesian Real-Time Optical Flow", Vision Interface 2002,
4. Liu, H., Hong, T.H., Herman, M., Camus, T.A., Chellappa, R., "Accuracy vs. Efficiency Trade-Offs in Optical Flow Algorithms", Computer Vision and Image Understanding, 72(3), 1998, pp. 271-286
5. P.C. Arribas, F.M.H. Macia, "FPGA Implementation of Camus Correlation Optical Flow Algorithm", Vision Interface 2001
6. M. Fleury, A.F. Clark and A.C.Downton, "Evaluating optical-flow algorithms on a parallel machine", Image and Vision Computing, 19(3), 2001, pp. 131-143.
7. Correia, M.V., Campilho, A.C., "Real-time implementation of an optical flow algorithm", International Conference on Pattern Recognition 2002, pp. 247-250.
8. H. Niitsuma and T. Maruyama, "Real-time Detection of Moving Objects", 14th International Conference on Field-Programmable Logic and Applications, 2004.
9. http://www.alpha-data.com
10. http://www.xilinx.com

Autonomous Operators for Direct Use on Irregular Image Data

S.A. Coleman[1] and B.W. Scotney[2]

[1] School of Computing and Intelligent Systems, University of Ulster, Northland Road, Londonderry, BT48 7JL, Northern Ireland
[2] School of Computing and Information Engineering, University of Ulster, Cromore Road, Coleraine, BT52 1SA, Northern Ireland
{sa.coleman, bw.scotney}@ulster.ac.uk

Abstract. Standard image processing algorithms for digital images require the availability of complete, and regularly sampled, image data. This means that irregular image data must undergo reconstruction to yield regular images to which the algorithms are then applied. The more successful image reconstruction techniques tend to be expensive to implement. Other simpler techniques, such as image interpolation, whilst cheaper, are usually not adequate to support subsequent reliable image processing. This paper presents a family of autonomous image processing operators constructed using the finite element framework that enable direct processing of irregular image data without the need for image reconstruction. The successful use of reduced data (as little as 10% of the original image) affords rapid, accurate, reliable, and computationally inexpensive image processing techniques.

1 Introduction

In image processing it is common to consider images as regular lattices of two-dimensional samples. However, irregularly sampled images can arise from motion or disparity compensation, such as in motion-compensated video coding, motion compensated video interpolation or disparity-compensated interpolation in stereoscopic images [9]. Irregularly spaced image data also occur frequently in areas such as remote sensing [11], medical imaging, oceanography and human retinal perception, [2, 3]. For example, in human retinal perception, human photoreceptors are not regularly distributed but have a personal signature denoted by the random positioning of cells. Hence, in order to replicate human retinal perception, use of irregular image data needs to be supported. Also, in underwater acoustic imaging systems, where an image is obtained by transmitting acoustic waves and sensing the waves reflected by objects using an array of sensors, various conditions may arise that lead to the array sensor data being irregular.

The application of standard image processing algorithms to irregularly sampled images often results in unreliable results. The underlying difficulty is that many such algorithms require the availability of complete, and regularly sampled, image data. This means that such algorithms are often applied to complete images that have been

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 296 – 303, 2005.

reconstructed from sparse irregularly sampled, and noisy data without a priori knowledge of the image content. Image reconstruction techniques include methods based on: cubic spline representation of the image [9]; wavelet representation of the image [10]; the theory of projections onto convex sets [8]. Other simpler techniques, such as low order image interpolation, whilst cheaper, are usually not adequate to support subsequent reliable image processing. Therefore a means of direct use of irregularly sampled images needs to be considered. Such an approach has a number of advantages including potential for efficient image coding [4].

In this paper we propose a family of autonomous image processing operators that can be applied directly to irregularly sampled images without the additional requirement of image reconstruction. These autonomous operators are able to change shape and size as required across the image plane through the use of adaptive Gaussian basis functions within the finite element framework. This approach has benefits with regard to reduced computational intensity and increased speed compared with methods that require full image reconstruction. In Sections 2 and 3 we introduce the finite element framework within which the autonomous operators are constructed. Section 4 presents the finite element implementation, which we call the *Primrose* algorithm, highlighting how the local operator size and shape are directly linked to the local neighbourhood point density. The efficient implementation of the operator design procedure is also discussed. In Section 5 we present the results of our algorithm using a little as 10% of the original image data.

2 Irregular Data Representation

We consider an irregularly sampled image to be represented by a spatially irregular sample of values of a continuous function *u(x,y)* of image intensity on a domain Ω. Our operator design procedure is then based on the use of a mesh generated using Delaunay triangulation [7].

With each node *i* in the mesh is associated a piecewise linear basis function $\phi_i(x, y)$ which has the properties

$$\phi_i(x_j, y_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

where (x_j, y_j) are the co-ordinates of the nodal point *j* in the mesh. Thus $\phi_i(x, y)$ is a "tent-shaped" function with support restricted to a small neighbourhood centred on node *i* consisting of only those triangles that have node *i* as a vertex; ϕ_i is linear on each mesh triangle. We then approximately represent the image function *u* by a function $U(x, y) = \sum_{j=1}^{N} U_j \phi_j(x, y)$ in which the parameters $\{U_1, ..., U_N\}$ are mapped from the image intensity values at the *N* irregularly located nodal, or scatter, points. The approximate image function representation is therefore piecewise linear on each triangle and has value U_j at node *j*.

3 Weak Form of Autonomous Operators

We formulate autonomous image operators that correspond to weak forms of operators in the finite element method [1, 5, 6]. Operators used for smoothing may be based simply on a weak form of the image function, for which it is assumed that the image function $u(x, y)$ belongs to the Hilbert space $H^0(\Omega)$; that is, the integral $\int_\Omega u^2 d\Omega$ is finite. Feature detection and enhancement operators are often based on first or second derivative approximations, for which it is necessary that the image function $u(x, y)$ is constrained to belong to the Hilbert space $H^1(\Omega)$; i.e. the integral $\int_\Omega (|\underline{\nabla} u|^2 + u^2) d\Omega$ is finite, where $\underline{\nabla} u$ is the vector $(\partial u/\partial x, \partial u/\partial y)^T$.

Corresponding to a first directional derivative $\partial u/\partial b \equiv \underline{b} \cdot \underline{\nabla} u$ or a second directional derivative $-\underline{\nabla} \cdot (\mathbf{B}\underline{\nabla} u)$, we may use a test function $v \in H^1(\Omega)$ to define the weak forms

$$E(u) = \int_\Omega \underline{b} \cdot \underline{\nabla} u v d\Omega \quad \text{and} \quad Z(u) = -\int_\Omega \underline{\nabla} \cdot (\mathbf{B}\underline{\nabla} u) v d\Omega .$$

Here $\mathbf{B} = \underline{b}\ \underline{b}^{\mathrm{T}}$ and $\underline{b} = (\cos\theta, \sin\theta)$ is the unit direction vector. Zero-crossing methods are often based on the isotropic form of the second order derivative, namely the Laplacian $-\underline{\nabla}.(\underline{\nabla} u)$. This is equivalent to the general form in which the matrix **B** is the identity matrix **I**.

In the finite element method a finite-dimensional subspace $S^h \subset H^1$ is used for function approximation; in our design procedure the irregular image U is a function in S^h, and S^h is defined by the irregular mesh of triangular elements and piecewise linear basis functions described in Section 2.

Since we are focusing on the development of autonomous operators that can explicitly embrace the concept of size and shape, our design procedure uses a finite-dimensional test space $T_\sigma^h \subset H^1$ that explicitly embodies a size parameter σ that is determined by the local scatter point density. This generalisation allows sets of test functions $\psi_i^\sigma(x, y)$, i=1,…,N, to be used when defining autonomous derivative based operators; for first and second order operators respectively, this provides the functionals

$$E_i^\sigma(U) = \int_\Omega \underline{b}_i \cdot \underline{\nabla} U \psi_i^\sigma d\Omega \text{ and } Z_i^\sigma(U) = \int_\Omega \underline{\nabla} U \cdot (\mathrm{B}_i \underline{\nabla} \psi_i^\sigma) d\Omega .$$

4 Design Procedure

The test space T_σ^h comprises a set of Gaussian basis functions $\psi_i^\sigma(x, y)$, i=1,…,N of the form

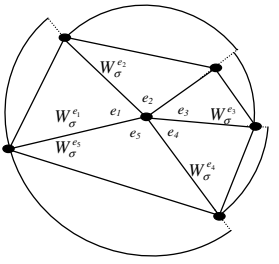
$$\psi_i^\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma^2}\right)}.$$

Each test function $\psi_i^\sigma(x, y)$ is restricted to have support over a neighbourhood Ω_i^σ, centred on node *i*, consisting of those triangular elements that have node *i* as a vertex. We note therefore that the integrals in the definitions of the functionals E_i^σ and Z_i^σ can be computed by integration over only the neighbourhood Ω_i^σ rather than the entire image domain Ω, providing the functionals

$$E_i^\sigma(U) = \int_{\Omega_i^\sigma} \underline{b}_i \cdot \underline{\nabla} U \psi_i^\sigma d\Omega_i \text{ and } Z_i^\sigma(U) = \int_{\Omega_i^\sigma} \underline{\nabla} U \cdot (\mathrm{B}_i \underline{\nabla} \psi_i^\sigma) d\Omega_i .$$

4.1 Autonomous Local Size Selection

The usually difficult issue of local size selection for an operator is now naturally addressed by the distribution of the nodes in the mesh-based representation of the image. For a scatter point (x_i, y_i) we obtain the local operator size directly from the mesh in the neighbourhood Ω_i^σ. We consider an approach which we have named the *Primrose* algorithm.



***Primrose* Operator**

Fig. 1. Neighbourhoods $\boldsymbol{\Omega_i^\sigma}$ in irregular mesh

In the *Primrose* algorithm the neighbourhood Ω_i^σ is defined to have a real-valued "radius" $W_\sigma^{e_m}$ for each element e_m in Ω_i^σ. In each case $W_\sigma^{e_m}$ is chosen as the radius of the smallest circle centred on (x_i , y_i) containing element e_m. Each element therefore contributes a “petal” to the *Primrose* operator as illustrated in Figure 1. The test function ψ_i^σ is correspondingly comprised of a set of sectors of Gaussian functions $\psi_i^{\sigma_m}$, where $\psi_i^{\sigma_m}$ is the test function over element e_m in Ω_i^σ. In each case choosing the element scale parameter $\sigma_m = W_\sigma^{e_m} / 1.96$ ensures that along the longest element edge of e_m through (x_i, y_i) 95% of the cross-section of the Gaussian is contained in e_m.

Construction of these autonomous operators on an irregular grid differs significantly from construction of image processing operators on a regular grid in that it is no longer appropriate to build explicitly an entire operator as in [6]; each operator throughout an irregular mesh is autonomous and may be different with respect to the operator neighbourhood size, shape, and the number of nodal points in the operator. When using an irregular grid, we work on an element-by-element basis to build each operator, taking advantage of the flexibility offered by the finite element method as a means of adaptively changing the irregular operator size and shape to encompass the data available in any local neighbourhood. Such local neighbourhoods are illustrated by the collections of triangular elements shown in Figure 2; in each neighbourhood the test function ψ_i^σ is comprised of a set of sectors of Gaussian functions $\psi_i^{\sigma_m}$ truncated at "radius" $W_\sigma^{e_m}$. Thus each operator is able to automatically alter its shape and size as required, dependent on the irregular node placement corresponding to the sampling of the image data. Operator *a* in Figure 2 has a central node with 5 adjoining nodes, operator *b* illustrates 7 adjoining nodes and operator *c* illustrates 6 adjoining nodes.

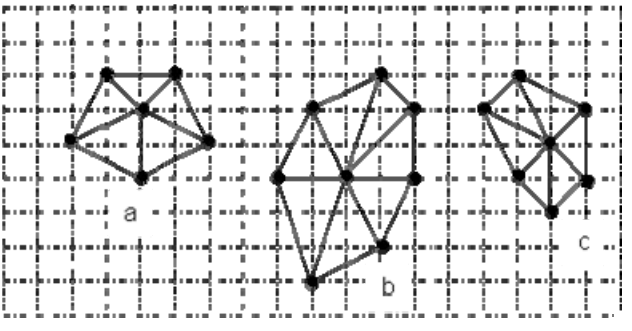


Fig. 2. Neighbourhoods of Autonomous Operators

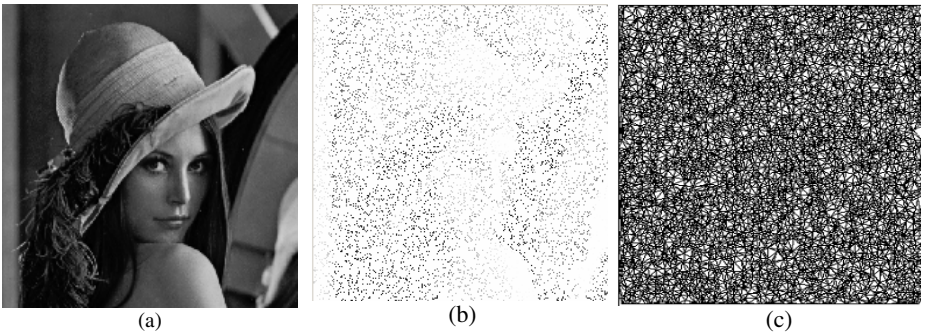
4.2 Efficient Implementation

The spatial relationship between the node *i* on which the Gaussian basis function ψ_i^σ is centred and each of the nodes in the neighbourhood Ω_i^σ is readily available from nodal numbering and locational information routinely stored in the finite element method. The integrals required in the computation of operators such as $E_i^\sigma(U)$ and $Z_i^\sigma(U)$ are evaluated through the standard process of finite element assembly: integrals $k_{is}^{m,\sigma}$ over each element $e_m \subset \Omega_i^\sigma$ are approximately evaluated using Gaussian quadrature rules, requiring function evaluations in e_m of the test function ψ_i^σ and of the three piecewise linear basis functions ϕ_s (locally indexed s = 1,2,3) whose support includes element e_m. Since the first derivatives of the piecewise linear basis functions are locally constant, each element integral may be accurately approximated by just four function evaluations (i.e., using a four-point Gauss rule).

The issue of evaluating integrals over irregularly shaped triangular elements is routinely handled in the finite element method by the use of isoparametric mappings that relate each element to a "standard" right-angled triangle on which numerical integration can be efficiently and accurately performed [1].

5 Results

To obtain results for our proposed technique, we simulated irregular image data by randomly selecting a specified proportion of an underlying 256×256 pixel intensity image. Such a sample is in no way based on *a prior* knowledge of the original image content. In order to apply our proposed technique to irregular image data, we initially use Delaunay triangulation to generate a triangular mesh in which the irregular image data points are nodes. As the nodal point set simply corresponds to the co-ordinates of the irregular image data, the local density of nodes in this mesh is simply controlled by the local availability of data points. This mesh is then used as the basis for the application of the finite element based autonomous operators described in Sections 3 and 4. Figure 3 illustrates an original 256×256 image and the corresponding randomly generated irregular image data and Delaunay triangulation.

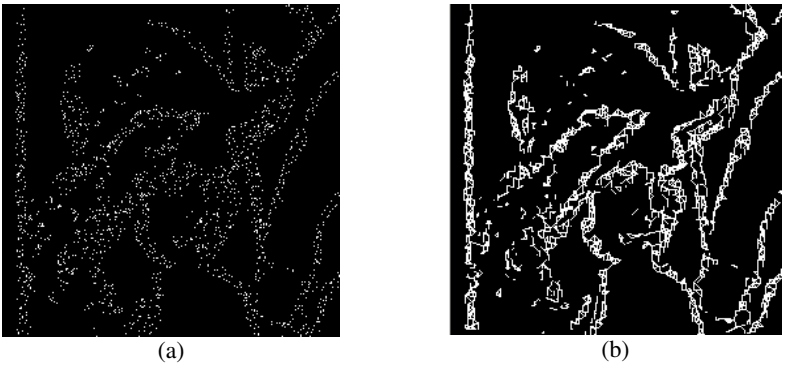


(a) (b) (c)

Fig. 3. (a) Original image (b) Irregular image data using a 10% sample (c) Delaunay triangulation

The autonomous operators are applied directly using the mesh illustrated in Figure 3(c) in order to generate a feature point set.

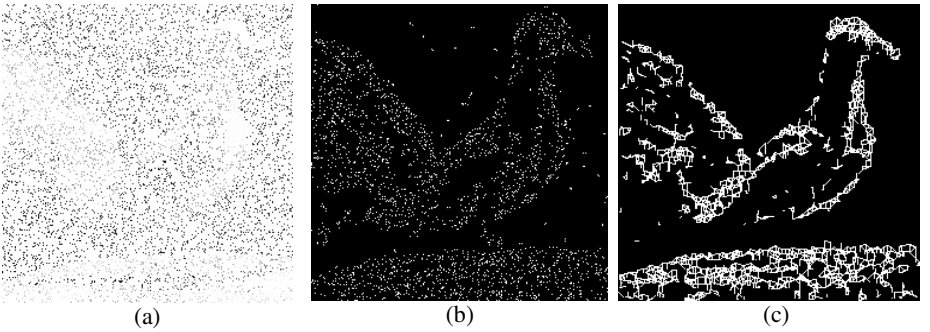
Figure 4(a) illustrates the thresholded feature point set generated when our family of autonomous operators corresponding to the first derivative operators $\{E_i^\sigma\}$ are applied to the triangular mesh illustrated in Figure 3(c). Already we can see the outline of the features appearing in this feature point set. To generate a more complete feature map, a simple edge linking technique was applied based on the similarity of edge direction between neighbouring feature points, and the results of this are illustrated in Figure 4(b). Although the feature map is not perfect, enough information is available to distinguish the main features in the irregular image data without the additional expense of image reconstruction to generate a regular data set.



(a) (b)

Fig. 4. (a) Feature point set generated from Figure 3(c); (b) Completed feature map

For further illustration, Figure 5 shows the same technique applied to 10% of another real image; again we can see that the main features appear quite clearly.



(a) (b) (c)

Fig. 5. (a) Irregular image data using a 10% sample; (b) Feature point set; (c) Completed feature map

6 Summary

In applications that rely on the use of incomplete image data, the processing of images to extract features is problematic without the availability of supplementary information on image content. This is because standard feature detection algorithms usually require the availability of complete and regularly sampled image data. Techniques that are based on image reconstruction without prior image knowledge do not generally provide reliable mechanisms for accurate feature extraction. In this paper we have presented a family of autonomous image processing operators based on a generalisation of the finite element method that can naturally formulate design procedures that can be used to successfully implement feature detection directly on non-uniformly sampled images.

Although this research currently uses a rudimentary edge linking technique, promising results have been presented, and further work will entail the application of the current operator design procedures to real images supported by more advanced edge linking algorithms. This problem will be explored for the dual purposes of feature extraction and for the further decomposition of the image into a very sparse content-based point set from which a reconstructed image may be cheaply yet accurately derived.

Acknowledgement

This work was funded by the Nuffield Foundation under the research grant number NAL/00839/G.

References

[1] Becker, E.B., Carey, G.F., and Oden, J.T., *Finite Elements: An Introduction*, Prentice Hall, London, 1981.

[2] Petrou, M., Piroddi, R., Chandra, S., "Irregularly Sampled Scenes" *Proceedings of SPIE Image and Signal Processing for Remote Sensing*, Vol., SPIE5573, 2004.

[3] Piroddi, R., Petrou, M., "Dealing with Irregular Samples" *Advances in Imaging and Electron Physics*, Vol.132, Elsevier, pp109-165, 2004.

[4] Ramponi, G., Carrato, S., "An Adaptive Irregular Sampling Algorithm and its Application to Image Coding" *Image and Vision Computing*, Vol.19 pp. 451-460, 2001.

[5] Scotney, B.W., Coleman, S.A., Herron, M.G., "A Systematic Design Procedure for Scalable Near-Circular Gaussian Operators." *Proc. IEEE ICIP,* pp 844-847, 2001.

[6] Scotney, B.W., Coleman, S.A., Herron, M.G., " Device Space Design for Efficient Scale-Space Edge Detection" *Proc. ICCS, Amsterdam,* LNCS 2329, Springer, pp1077-1086, 2002.

[7] Shewchuk, J.R., "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator" *1st Workshop on Applied Computational Geometry*, pp. 124-133, 1996

[8] Stasinski, R., Konrad, J., "POCS-Based Image Reconstruction from Irregularly-Spaced Samples" *Proceedings of IEEE International Conference on Image Processing*, pp. 315-318, 2000.

[9] Vazquz, C., Dubois, E., Konrad, J., "Reconstruction of Irregularly-Sampled Images by Regularization in Spline Spaces" *Proceedings of IEEE International Conference on Image Processing*, pp. 405-408, 2002.

[10] Vazquz, C., Konrad, J.,Dubois, E., "Wavelet-Based Reconstruction of Irregularly-Sampled Images : Application to Stereo Imaging" *Proceedings of IEEE International Conference on Image Processing*, pp. 319-322, 2000.

[11] Yegnanarayana B, Mariadassou CP & Saini P. "Signal Reconstruction from Partial Data for Sensor Array Imaging applications", Signal Processing, Vol. 19, pp.139-149, 1990.

Texture Granularities

Paul Southam and Richard Harvey

University of East Anglia, Norwich,
Norfolk, NR4 7TJ, England

Abstract. We introduce three new texture features that are based on the morphological scale-space operator known as the sieve. The new features are tested on two databases. The first, the Outex texture database, contains Brodatz-like textures captured under constant illumination, scale and rotation. The second, the Outex natural scene database, contains images of real-world scenes taken under variable conditions. The new features are compared to univariate granulometries, with which they have some similarities, and to the dual-tree complex wavelet transform, local binary patterns and co-occurrence matrices. The features based upon the sieve are shown to have the best overall performance.

1 Introduction

Granulometries [10] have had a long history in texture analysis. They have been used for the analysis of digital mammograms [2], radiographic imaging of lungs [17] and Diatom classification [16]. Univariate granulometries comprise varying-scale morphological openings and closings applied in parallel using a fixed shape structuring element, scaled by a parameter. It is known [1], that the shape of the structuring element affects texture classification. This poses in interesting question, is it the shape of the structuring element that is important or the analysis over scale? Until recently it was not possible to separate these two criteria because the shape of the filter is fixed. However, by using a different class of mathematical morphology filters called sieves, it is possible to analyse an image by scale without the filter imposing a shape – a technique we explore in this paper.

First we justify our evaluation methods and choose from the many texture databases available. Next we introduce sieves and show how they can provide new texture filters. In the final sections we evaluate the performance of these systems and compare them to a number of benchmark systems including granulometries.

2 Databases

This paper uses Outex [11] which has recently become regarded as the best available framework for evaluating texture [15]. It contains several tasks. Here we use TC_00000 which is the texture classification task using textures imaged under consistent conditions (perpendicular to the surface, fixed scale, rotation and illumination and so on). Examples of these textures are shown in Figure 2. Stylised

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 304–311, 2005.

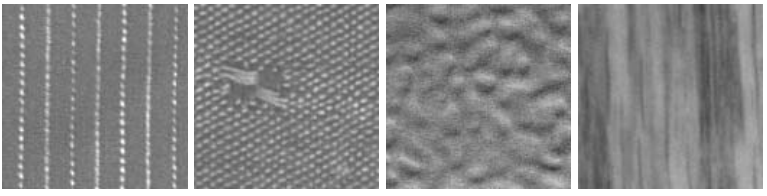


Fig. 1. Various Outex samples used in Outex task TC_00000

Fig. 2. Various Outex natural scene images (left) and hand-segmented ground-truth (right)

textures such as these are extremely common in the literature (Brodatz[6] and MeasTex [14] are examples) and automatic texture classifiers are known to perform extremely well on such data ([9,1] for example). Unfortunately, so far, the performance of such systems has not extrapolated to real-world scenes ([8] and [13]). The natural scenes Outex database contains 20 colour images (2272×1704 pixels) of real-world scenes taken with a digital camera under varying illumination and orientation. The view is said to be "roughly consistent, simulating a navigating vehicle" [13]. There five defined texture classes, *sky*, *trees*, *grass*, *road* and *buildings* which are defined through hand-labelled regions in ground-truth images. Examples of the scene and ground truth images are shown in Figure 2.

3 Sieve

Sieves are described as a one-dimensional non-linear scale-space decomposition algorithm in [5] and are extended to n-dimensions in [3] by adopting techniques from graph morphology. Sieves use morphological scale-space operators, specifically openings and closings, or combinations of them, to filter an input signal by removing extrema of specific scale. They apply flat structuring elements to an input signal which, unlike conventional morphological operators such as those used in granulometries, have a fixed size but varying shape. A stated advantage of this approach is that the shape of the structuring element is not visible in filtered signal.

The sieve performs a decomposition by scale via the structure shown in Figure 3. At each stage the filtering operator φ removes extrema of only that scale.

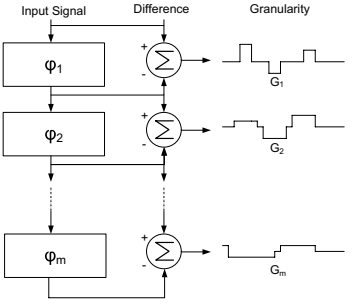


Fig. 3. The structure of a 1D sieve decomposition where φ is a filtering operator chosen from a set [5,3]. Non-zero regions in the output are called *granules* and the set of granules is called the *granularity domain* in an analogy to granulometries.

At the first stage φ_1 removes extrema of scale 1 (removed extema are called *granules*), φ_2 removes extrema of scale 2 and so on until the maximum scale m (which is the number of pixels in the image). This serial structure can be contrasted with the parallel structure used in granulometries. Because objects in images are often delineated by iso-intensity contours, sieves have been applied to image segmentation tasks in which semantically meaningful objects are removed at a specific (typically higher) scale. At lower scales the sieve can been seen to remove at first image noise then textural information. Figure 4 shows a example sieve decomposition of an image using a 2D M-filter sieve.

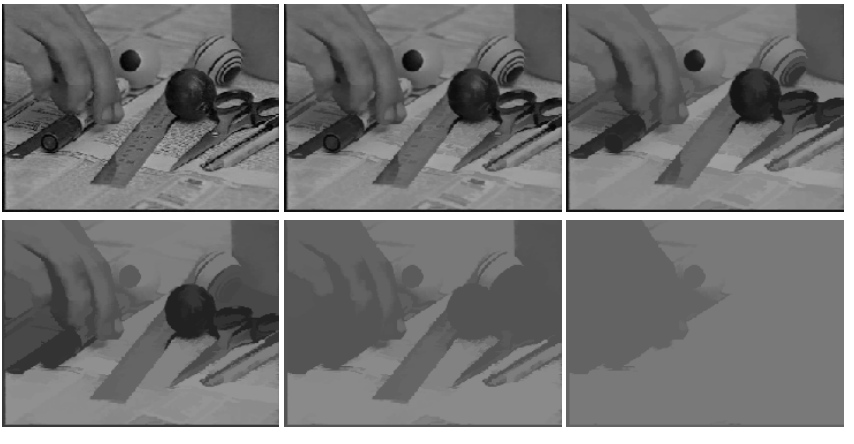


Fig. 4. An original image (top left) sieved to scales, reading left-to-right, top-to-bottom, 15, 251, 2500, 8500 and 25000 . Each image has fewer intensity extrema than its predecessor. A full decomposition may be summed to re-create the original thus the sieve is a transform of the original image.

4 Methods

Here we introduce three new sieve-based texture features. All the new methods are based on granules. In [3], granule images are defined as the difference between successive sieve outputs, $G_n = \mathcal{S}_n - \mathcal{S}_{n-1}$ where $\mathcal{S}_n$ is the nth stage in the serial structure shown in Figure 3. There are thus a great number of granule images. The G_n are a transform of the texture which can be reconstructed through a simple summation. Here, each texture image is sieved to a few scales, $[s_1 \dots s_N]$ where $\log_{10} s_n$ are equispaced between 0 and $\log_{10} P$, where $N = 5$ and $P = 30$ are chosen to remove all textural information from all images. The difference between these images are termed *channels*, $C_n = \mathcal{S}_{s_n} - \mathcal{S}_{s_{n-1}}$. The intensity of the granule, or channel, images as a function of scale is an indicator of the scale-distribution of the texture features. The mean, standard deviation and skewness of the magnitude of the granule images make suitable features.

In the first method φ is a 2D M-filter [3] which filters the image using an morphological opening followed by a morphological closing in one operation. This produces bi-polar granule images that are invariant to simple rotation of the texture image.

In the second method φ is a 1D recursive median filter [4] where each image is sieved at orientations of $\pm 30°, \pm 60°, \pm 90°$ which produces bi-polar granule images which are sensitive to the image rotation.

The final method (*oc*-sieve) is similar to granulometries because it uses two sieves: one using an opening filter (*o*-sieve) and the other using a closing filter (*c*-sieve). This produces twice the number of granule images as in the first method, one set of positive granule images derived from only the image maxima and another set derived from only the image minima. The results from this method highlights the difference between processing maxima and minima in on one bi-polar operation as in the 2D M-sieve, and separately using the uni-polar *o*-sieve and *c*-sieve.

For comparison we also implement features based on, the dual-tree complex wavelet transform [7], Local binary pattern analysis [12], co-occurrence matrices, and three granulometic methods with different sized structuring elements. The first granulometic method uses a combination of disc and vertical structuring elements (denoted GDV subsequently) as these are known to perform best on a selection of Brodatz textures [1]. We also implement a rotation-invariant granulometic method based upon disc and square structuring elements (denoted GDS) for comparison with the 2D sieve; and a rotation-sensitive method using line structuring elements at angles of $\pm 30°, \pm 60°, \pm 90°$ (denoted GLA) for comparison with the 1D sieve. In all the granulometic methods the size of the structuring element is within the scales used for a sieve channel. Features are formed from the means, standard deviations and skewnesses of the differences between successive openings and closings of the image as with the *oc*-sieve method.

In all the methods the results are always improved by re-projecting the feature vectors via principal component analysis (PCA) with the further possibility to reduce feature dimension. During classification the Euclidean distances are measured between each feature vector in the PCA space and a k nearest neigh-

bour classifier is used to predict the test class. For test set TC_00000 it is known that $k = 1$ is optimal [15] (indicating, incidentally, that TC_00000 could be improved with more data). For the natural scenes data we find $k = 3$ to be the best.

5 Results

Outex_TC_00000 comprises 480 images (24 texture classes each with 20 images). There are 100 standard learning tasks. Each has 240 training and 240 test images. Table 1 shows the mean success rate over the 100 tasks using the nearest neighbour classifier with a Euclidean distance measure on the feature vector after PCA (the final column of Table 1 (left) is the size of the PCA vector without truncation). Note that the success rate of the LBP method differs from [12] because, for consistency, we are not using their histogram distance measure.

These results show (as in [1]) that oriented structuring elements (GLA and GDV) are the best among the granulometric methods. However both the DTCWT and 1D sieve score better in this test and are the best performing overall. The DTCWT produces eight sub-bands per level so three levels produce $8 + 8^2 + 8^3$ bands hence 336 features (mean and standard deviation of the absolute value). The 1D-sieve feature (and the GLA feature) is formed over only five channels (at scales 0 to 1, 1 to 2, 2 to 5, 5 to 13 and 13 to 30) and six orientations which is only 30 components so we can afford to compute the mean, standard deviation and skewness which still retaining fewer features than the DTCWT. The 2D- and oc-sieves have channels of the same scales as the GLA and 1D-sieve. The 2D-sieve hence as 15 features (mean, standard deviation and skewness per channel) and the oc-sieve has double the number because it has channels at positive and negative scales. LBP has 256 features as described in [12]. The co-occurrence matrices have 12 features representing energy, inertia, entropy and homogeneity at orientations of $0°, 90°$ and $45°$. GDV has 48 features which are the mean, standard deviation and skewness over the five standard channels for the vertical element. The circular element structuring elements must have integer radii which leads us to choose only three scales for the discs (radii 1, 2 and 3). GDS uses the same discs and squares with sides of 1,2,4 and 5 pixels.

Table 1 (right) shows the result of running McNemar's test at a significance of $\alpha = 0.05$. The entries show the number of times, out of a 100 runs, that the test allows us to reject the null hypothesis. Ignoring any arguments about Bonferroni adjustments, this is a crude measure of whether the texture classifiers differ significantly. The 1D-sieve, DTCWT, GDV and GLA have very similar performance with 1D-sieve and DTCWT the best performing. Using PCA to reduce feature dimensionality improves the 1D sieve success rate to 0.999 which is the same as the DTCWT but using only 40 features. The effect of applying PCA to the DTCWT incurs no performance increase but maintains a 0.999 success rate using 77 features.

For the Outex natural scene database, features are generated from each hand-segmented region in the 20 images in this database. Not all texture methods are

Table 1. Left: $\bar{x} = 1$ - e (success rate), max and min success rate out of the 100 trials, standard deviation σ, (for standard error divide by ten) and number of features f for Outex test suite Outex_TC_00000. **Right:** The number of times out of the Outex_TC_00000 100 trials that, under McNemar's test, we can confidently ($\alpha = 0.05$) reject the null hypothesis that the two data distributions are drawn from the same source.

	$\bar{x}$	max	min	σ	f
1D sieve	0.998	1	0.988	0.0034	90
GLA	0.995	1	0.975	0.0054	90
2D sieve	0.954	0.975	0.929	0.0106	15
GDS	0.986	1	0.954	0.0081	42
GDV	0.995	1	0.975	0.0046	48
oc-sieve	0.970	0.988	0.938	0.0105	30
DTCWT	0.999	1	0.988	0.0019	336
LBP	0.986	0.996	0.967	0.0007	256
co-occ	0.946	0.983	0.900	0.0015	12

	1D Sieve	GLA	2D sieve	GDS	sieve OC	DTCWT	LBP	Co-occ	GDV
1D sieve	0	0	95	8	61	0	10	96	0
GLA	-	0	88	3	39	0	3	94	0
2D sieve	-	-	0	54	14	98	43	2	87
GDS	-	-	-	0	16	11	1	71	2
sieve OC	-	-	-	-	0	64	8	25	82
DTCWT	-	-	-	-	-	0	12	96	0
LBP	-	-	-	-	-	-	0	67	3
Co-occ	-	-	-	-	-	-	-	0	94
GDV	-	-	-	-	-	-	-	-	0

easily applicable to this set because the regions are hand-drawn so, for example, wavelet support-regions are not guaranteed to fit the ground-truth. We therefore restrict the comparison to sieves and granulometries because, for these, we can generate the filtered images, apply the hand-segmented region as a mask, and generate a feature for each region. There are a total of 91 labelled regions which is too few for holdout. Therefore we use leave-out-one cross-validation. Table 2 (left) shows the success rate across all classes, with a knn classifier ($k = 3$, Euclidean distance). Also shown is the number of samples per class. GLA and GDV perform quite well as expected. However it appears that both are quite fragile – certain types of texture are difficult to classify with these methods. The best overall performers are the oc- and 2D-sieves.

Table 2 (right) shows the p-values obtained under McNemars test where $p > 3.84$ allows the null hypothesis (that the two classifiers are indistinguishable) to be rejected at $\alpha = 0.05$ with oc-sieve probably the best performing overall.

6 Conclusions and Discussion

The Outex_TS_00000 test suite has images of texture at a large scale captured under consistent illumination and rotation. This test set is highly representative of the data used to evaluate the majority of texture classifiers over the past thirty years. Scholarly interest in granulometries seems to have declined recently but here we show that granulometries are among the best-performing particularly when using directional structuring elements. The overall trend in these results is

Table 2. Left: Number of samples per class, mean success rate for each class and overall mean success rate for the Outex natrual scene database. **Right:** McNemar's p-values ($p > 3.84$ is the threshold for rejecting the null hypothesis at $\alpha = 0.05$)

	No. Samples	1D Sieve	GLA	2D sieve	GDS	GDV	sieve OC
sky	14	0.79	1	1	0.93	1	1
tree	17	0.59	0.76	0.82	0.76	0.71	0.88
bush	15	0.20	0.27	0.53	0.60	0.33	0.53
grass	20	0.60	0.70	0.65	0.70	0.75	0.80
road	16	0.38	0.56	0.88	0.56	0.63	0.69
building	9	0	0.22	0.33	0.11	0.33	0.56
mean		0.42	0.59	0.70	0.60	0.62	0.74

	1D Sieve	GLA	2D sieve	GDS	sieve OC	GDV
1D sieve	0	5.63	14.69	7.03	19.31	8.65
GLA	-	0	3.12	0.05	5.33	0.10
2D sieve	-	-	0	2.04	0.31	1.75
GDS	-	-	-	0	3.70	0.06
sieve OC	-	-	-	-	0	3.70
GDV	-	-	-	-	-	0

that rotationally invariant methods such as GDS, the oc-sieve and the 2D-sieve perform poorly, implying that, when the orientation of the texture is known, then there is no advantage to rotationally invariant features [15]. The DTCWT is also directionally sensitive and performs well but it uses a large number of features compared to the 1D-sieve.

When the rotation of the texture is unknown, a more realistic situation for unconstrained computer vision, then directional texture features are likely to perform poorly. For this reason one seeks systems that are able to operate on natural scenes. In the Outex natural scene database the 2D-sieves and oc-sieves have the highest success rates (and are usually the best performing on individual classes) followed by the GDV and GDS. The 1D-sieve, which was one of the best performing methods on Outex_TS_00000, is now the worst which is a concern for those who wish to extrapolate from the performance on stylised texture images to reality.

References

1. G. Ayala and J. Domingo. Spatial size distributions: applications to shape and texture anaylsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1430–1442, 2001.
2. S. Baeg, S. Batman, E. Dougherty, V. Kamat, N. Kehtarnavaz, S. Kim, A. Popov, K. Sivakumar, and R. Shah. Unsupervised morphological granulometric texture segmentation of digital mammograms. *Electronic Imaging*, 8(1):65–75, 1999.
3. J. Bangham, R. Harvey, P. Ling, and R. Aldridge. Morphological scale-space preserving transforms in many dimensions. *Journal of Electronic Imaging*, 5:283–299, 1996.
4. J. Bangham, P. Ling, and R. Young. Multiscale recursive medians, scale-space, and transforms with applications to image-processing. *IEEE Trans Image Processing*, 5(6):1043–1048, June 1996.

5. J. A. Bangham, P. Chardaire, C. J. Pye, and P. D. Ling. Multiscale nonlinear decomposition: The sieve decomposition theorem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):529–539, May 1996.
6. P. Brodatz. *Textures : A Photographic Album*. New York : Dover, 1966.
7. S. Hatipoglu, K. Mitra, and N. Kingsbury. Texture classification using dual-tree complex wavelet transform. In *Image Processing and its Applications*, pages 344–347. IEE, 1999.
8. P. Howarth and S. Ruger. Evaluation of texture features for content-based image retrieval. In *International Conference on Image and Video Retrieval (CIVR) 2004*, pages 326–324, 2004.
9. B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.
10. G. Matheron. *Random sets and integral geometry*. John Wiley, New York, 1975.
11. T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. In *Proc. 16th International Conference on Pattern Recognition, Quebec, Canada*, volume 1, pages 701–706, 2002.
12. T. Ojala, M. Pietikanien, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.
13. M. Pietikainen, T. Nurmela, T. Maenpaa, and M. Turtinen. View-based recognition of real-world textures. *Pattern Recognition*, 37:313–323, 2004.
14. G. Smith and I. Burns. Measuring texture classification algorithms. *Pattern Recognition Letters*, 18:1495–1501, 1997.
15. P. Southam and R. Harvey. Compact rotation-invariant texture classification. In *International Conference on Image Processing (ICIP 2004)*, pages 3033–3036. IEEE, 24–27 Oct 2004.
16. E. Urbach, J. Roerdink, and M. Wilkinson. Connected rotation-invariant size-shape granulometries. In *17th International Confernece on Pattern Recognition (ICPR) 2004*,, volume 1, pages 688–691, 2004.
17. M. Vanrell and J. Vitria. Mathematical morphology, granulometries, and texture perception. *Image algebra and morphological image processing IV*, SPIE-2030:152–161, 1993.

Enhancement of Noisy Images with Sliding Discrete Cosine Transform

Vitaly Kober[1] and Erika Margarita Ramos Michel[2]

[1] Department of Computer Science, CICESE, Ensenada, B.C. , Mexico
vkober@cicese.mx
[2] University of Colima, Colima, Mexico
ramem@ucol.mx

Abstract. Enhancement of noisy images using a sliding discrete cosine transform (DCT) is proposed. A minimum mean-square error estimator in the domain of a sliding DCT for noise removal is derived. This estimator is based on a fast inverse sliding DCT transform. Local contrast enhancement is performed by nonlinear modification of denoised local DCT coefficients. To provide image processing in real time, a fast recursive algorithm for computing the sliding DCT is utilized. The algorithm is based on a recursive relationship between three subsequent local DCT spectra. Computer simulation results using a real image are provided and discussed.

1 Introduction

Many different image enhancement techniques have been introduced to improve the visual appearance of images [1-7]. These techniques may be broadly divided in two classes. The first class is based on decomposing an image onto high- and low-frequency signals, manipulating them separately and then combining them. Examples of such methods are homomorphic filtering [1] and unsharp masking [2]. The second class consists of various histogram modification techniques [3]. The classical unsharp masking is one of the most commonly used methods for image enhancement because it works well in many real applications. In this method a fraction of the high-frequency signal of an image is added to the original image itself to form a locally enhanced image. Drawbacks of the unsharp masking are as follows. A linear highpass filter makes the system very sensitive to noise. This results in undesirable noise enhancement in flat and high-contrast areas of even slightly noisy images. The operation also uses a constant scaling factor that may lead to overshoot artifacts in high-contract areas of the image. Various methods have been proposed to improve the performance of the unsharp masking [4-6]. The use of quadratic filters instead of a linear highpass filter enhances details and edges in accordance with a human perceptual criterion. These filters can be described as local mean weighted highpass filters. Weighting the highpass filter output by the local mean value leads to enhancement in dark areas less than that of in bright areas. This coincides with Weber's law, which states that the just noticeable brightness difference is proportional to average background brightness. Consequently the perceived noise is reduced

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 312 – 319, 2005.

comparing with that of the unsharp masking output. These methods have all proven to be effective in local enhancing of images. However, the methods discussed above use a moving window with fixed size and shape (usually 3x3 pixels). This may not be equally effective in enhancing various structures in an image, which can vary widely in size and shape. Examples such structures are fine edges and middle-sized details. Moreover, these structures can be degraded due to different kinds of noise. Obviously that additive noise could be better suppressed using a large moving window. Thus there is a need to design a technique, which preserves and enhances different-sized structures in the original image, while eliminating noise. Methods using a spatially adaptive filter mask (neighborhood), whose size and shape are adjustable to local details to be processed, are appropriate for this purpose. The price for the adaptivity to the local signal is a high computational complexity. Recently, some algorithms with adaptive filter masks for image enhancement were proposed [7]. A highpass filtering is considered as an enhanced difference between the central pixel of a moving window and a smoothed version of the original image. The smoothing is performed with rank-order filters over various adaptive neighborhoods defined around the central pixel of a moving window.

In this paper, we carry out enhancement of noisy images using a sliding discrete cosine transform (DCT) coefficients. The sliding DCT is based on the concept of short-time signal processing [8]. The short-time orthogonal transform of a signal x_k is defined as

$$X_s^k = \sum_{n=-\infty}^{\infty} x_{k+n} w_n \psi(n,s), \tag{1}$$

where w_n is a window sequence, $\psi(n,s)$ represents the basis functions of an orthogonal transform. In this paper we use one-dimensional notation for simplicity. Equation (1) can be interpreted as the orthogonal transform of x_{k+n} as viewed through the window w_n. X_s^k displays the orthogonal transform characteristics of the signal around time k. Note that while increased window length and resolution are typically beneficial in the spectral analysis of stationary data, for time-varying data it is preferable to keep the window length sufficiently short so that the signal is approximately stationary over the window duration. It is of interest to note that there is a link between wavelet and sliding short-time transforms. Wavelet transforms [9] are signal sub-band decompositions by filters with frequency responses formed from the Fourier transform of a mother function on a given scale. On the other hand, short-time signal transformation is a signal sub-band formed by a filter with frequency response equal to the Fourier transform of the windowed function of used orthogonal transform. The main distinction between these two sub-band decompositions is that the short-time transform decomposes signal into sub-bands of the same width uniformly arranged in the base, while sub-bands of the wavelet are arranged in logarithm scale and have width that is doubled with the increase of its frequency. Now we assume that the window has finite length around n=0, and it is unity for all $n \in [-N_1, N_2]$. Here N_1 and N_2 are integer values. This leads to signal processing in a sliding window [10]. In other words, local filters in the domain of an orthogonal transform at each position of a moving window modify the orthogonal transform coefficients of a signal to obtain only an estimate of the pixel x_k of the window. The choice of orthogonal transform for

sliding signal processing depends on many factors. The DCT is one the most appropriate transform with respect to the accuracy of power spectrum estimation from the observed data that is required for local filtering, the filter design, and computational complexity of the filter implementation. Linear filtering in the domain of DCT followed by inverse transforming is superior to that of the discrete Fourier transform (DFT) because a DCT can be considered as the DFT of a signal evenly extended outside its edges. This consequently attenuates boundary effects caused by circular convolution that are typical for linear filtering in the domain of DFT.

The proposed algorithm at each position of a moving window consists of two steps: first, noise suppression is performed in the domain of a sliding DCT, and then local contrast enhancement is carried out with nonlinear modification of denoised local DCT coefficients. The size of a moving window determines the maximum size of details to be enhanced. The presentation is organized as follows. In Section 2, we review recursive algorithms for computing the sliding forward and inverse DCTs. In Section 3, a local adaptive filter minimizing the minimum mean-square error defined in the domain of the sliding DCT is derived. In section 4, we test the filter performance to enhance a real aerial image. Section 5 summarizes our conclusions.

2 Fast Forward and Inverse Algorithms of Sliding DCT

The discrete cosine transform is widely used in many signal processing applications such as adaptive filtering, video signal processing, feature extraction, and data compression. This is because the DCT performs close to the optimum Karhunen-Loeve transform for the first-order Markov stationary data, when the correlation coefficient is near 0.9 [11]. Recently, fast forward and inverse algorithms for fast computing of DCTs were proposed [12]. The sliding cosine transform (SCT) is defined as

$$X_s^k = \sum_{n=-N_1}^{N_2} x_{k+n} \cos\left(\pi \frac{\left(n+N_1+1/2\right)s}{N}\right), \tag{2}$$

where $N=N_1+N_2+1$, $\{ X_s^k ; s=0, 1,..., N-1\}$ are the transform coefficients around time k. The coefficients of the DCT can be obtained as $\{ C_0^k = X_0^k / \sqrt{2} ; C_s^k = X_s^k , s=1,..., N-1\}$. The SCT on the base of a recursive relationship between three subsequent local DCT spectra [12] is given by

$$X_s^{k+1} = 2X_s^k \cos\left(\frac{\pi s}{N}\right) - X_s^{k-1} + \cos\left(\frac{\pi s}{2N}\right)\left(x_{k-N_1-1} - x_{k-N_1} + (-1)^s \left(x_{k+N_2+1} - x_{k+N_2}\right)\right). \tag{3}$$

We see that the computation of the DCT at the window position $k+1$ involves values of the input sequence x_k as well as the DCT coefficients computed in two previous positions of the moving window.

Table 1. Number of arithmetical operations for computing of sliding DCT

	Number of additions	Number of multiplications
Fast DCTs [13]	3MN/2 - N+1	MN/2+1
Recursive algorithm	2N+5	2N-1

Tables 1 provides a comparison of the computational complexity of the recursive algorithm with fast DCT algorithms. The length of a moving window for the recursive algorithm is an arbitrary integer value determined by characteristics of a signal to be processed. In contrast, fast DCT algorithms require the length to be of a power of 2, $N=2^M$. If x_k is the central pixel of the window, that is, $N_1=N_2$ and $N=2N_1+1$, then the inverse transform is written as

$$x_k = \frac{1}{N}\left(2\sum_{s=1}^{N_1}(-1)^s X_{2s}^k + X_0^k\right). \tag{4}$$

We note that in the computation only the spectral coefficients with even indices are involved. The computation requires one multiplication and N_1+1 additions.

3 Signal Denoising in the Domain of Sliding DCT

First we define a local criterion of the performance of filters for image and signal processing and then derive optimal local adaptive filters with respect to the criterion. One the most used criterion in signal processing is the minimum mean-square error (MMSE). Since the processing is carried out in a moving window, then for each position of a moving window an estimate of the central element of the window is computed. Suppose that the signal to be processed is approximately stationary within the window. The signal may be distorted by sensor's noise.

Let us consider a generalized linear filtering of a fragment of input one-dimensional signal (for instance for a fixed position of the moving window). Let $\boldsymbol{a}=[a_k]$ be undistorted real signal, $\boldsymbol{x}=[x_k]$ be observed signal, $k=1,\ldots, N$, N be the size of the fragment, $\boldsymbol{U}$ be the matrix of the discrete cosine transform, $E\{.\}$ be the expected value, superscript ^{T} denotes the transpose. Let $\bar{\boldsymbol{a}} = \boldsymbol{H}\boldsymbol{x}$ be a linear estimate of the undistorted signal, which minimizes the MMSE averaged over the window

$$MMSE = E\left\{(\boldsymbol{a}-\bar{\boldsymbol{a}})^T(\boldsymbol{a}-\bar{\boldsymbol{a}})\right\}\Big/N. \tag{5}$$

The optimal filter for this problem is the Wiener filter [11]:

$$\boldsymbol{H} = E\left\{\boldsymbol{a}\,\boldsymbol{x}^T\right\}\left[E\left\{\boldsymbol{x}\,\boldsymbol{x}^T\right\}\right]^{-1}. \tag{6}$$

Let us consider the known model of signal:

$$x_k = \sum_n w_{k,n}a_n + v_k, \tag{7}$$

where $\boldsymbol{W}=[w_{k,n}]$ is a distortion matrix, $\boldsymbol{v}=[v_k]$ is additive noise with zero mean, $k,n=1,\ldots N$, N is the size of fragment. The equation can be rewritten as

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{a} + \boldsymbol{v}, \tag{8}$$

and the optimal filter is given by

$$\boldsymbol{H} = \boldsymbol{K}_{aa}\boldsymbol{W}^T\left[\boldsymbol{W}\boldsymbol{K}_{aa}\boldsymbol{W}^T + \boldsymbol{K}_{vv}\right]^{-1}, \tag{9}$$

where $\boldsymbol{K}_{aa} = E\left\{\boldsymbol{a}\boldsymbol{a}^T\right\}$, $\boldsymbol{K}_{vv} = E\left\{\boldsymbol{v}\boldsymbol{v}^T\right\}$, $E\left\{\boldsymbol{a}\,\boldsymbol{v}^T\right\} = 0$ are the covariance matrices. It is assumed that an input signal and noise are uncorrelated.

The obtained optimal filter is based on an assumption that an input signal within the window is stationary. The result of filtering is the restored window signal. This corresponds to signal processing in non-overlapping fragments. Now suppose that the signal is processed in a moving window in the domain of the sliding DCT. For each position of the window an estimate of the central pixel should be computed. Using the equation for inverse sliding DCT presented in the previous section, the point-wise MSE for reconstruction of the central element of the window can be written as follows:

$$PMSE(k) = E\left\{\left[a(k) - \overline{a}(k)\right]^2\right\} = E\left\{\left[\sum_{l=1}^{N} \alpha(l)\left(A(l) - \overline{A}(l)\right)\right]^2\right\}, \qquad (10)$$

where $\overline{\boldsymbol{A}} = \left[\overline{A}(l) = H(l)\,X(l)\right]$ is a vector of signal estimate in the domain of the DCT, $\boldsymbol{H}_U = \left[H(l)\right]$ is a diagonal matrix of the scalar filter, $\boldsymbol{\alpha} = \left[\alpha(l)\right]$ is a diagonal matrix of the coefficients of inverse sliding cosine transform (4). Minimizing (10), we obtain

$$\boldsymbol{H}_U = \left[\boldsymbol{P}_{xx}\right]^{-1} \boldsymbol{P}_{\boldsymbol{\alpha x}}\, \boldsymbol{I}_{\alpha}. \qquad (11)$$

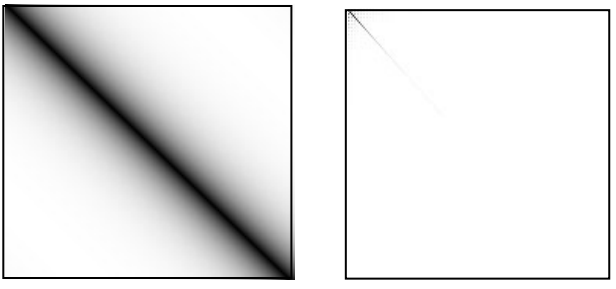
where $\boldsymbol{P}_{\boldsymbol{\alpha x}} = \left[E\left\{A(l)\,X(k)\right\}\right]$, $\boldsymbol{P}_{xx} = \left[E\left\{X(l)\,X(k)\right\}\right]$, $\boldsymbol{I}_{\boldsymbol{\alpha}}$ is the identity matrix of the dimension of $\boldsymbol{\alpha}$. Note that matrix of coefficients $\boldsymbol{\alpha} = \left[\alpha(l)\right]$ for the inverse sliding transform (4) is singular. The inverse sliding cosine transform (4) possesses the dimension of the matrix twice less than the size of the window signal. Therefore, the computational complexity of the scalar filters in (11) and signal processing can be significantly reduced comparing to the complexity for the filter in (6). For the model of signal distortion in (8) the filter matrix is given as

$$\boldsymbol{H}_U = \left[\boldsymbol{U}\left(\boldsymbol{W}\boldsymbol{K}_{aa}\boldsymbol{W}^T + \boldsymbol{K}_{vv}\right)\boldsymbol{U}^T\right]^{-1} \boldsymbol{U}\,\boldsymbol{K}_{aa}\boldsymbol{W}^T\boldsymbol{U}^T\,\boldsymbol{I}_{\alpha}. \qquad (12)$$

If a signal has a high correlation coefficient and a smoothed version of the signal is corrupted by additive, weakly-correlated noise, then the matrix $\boldsymbol{U}\left(\boldsymbol{W}\boldsymbol{K}_{aa}\boldsymbol{W}^T + \boldsymbol{K}_{vv}\right)\boldsymbol{U}^T$ in (12) is close to diagonal. Figure 1 shows the covariance matrix of a smoothed, noisy, one-dimensional signal having the correlation coefficient of 0.95 as well as the discrete cosine transform of the covariance matrix. The linear convolution between a signal $\boldsymbol{x}$ and the matrix $\boldsymbol{K}_{aa}\boldsymbol{W}^T$ in the domain of the sliding DCT can be well approximated by a diagonal matrix $Diag\left(\boldsymbol{U}\boldsymbol{K}_{aa}\boldsymbol{W}^T\boldsymbol{U}^T\,\boldsymbol{I}_{\alpha}\right)\boldsymbol{X}$. Therefore, the matrix of the scalar filter in (12) is close to diagonal, and the filter can be written as

$$H(l) \approx \frac{P_1(l)}{P_2(l) + P_{vv}(l)}, \qquad (13)$$

where $P_1(l)$, $P_2(l)$, $P_{nn}(l)$ are diagonal elements of the following matrices $\boldsymbol{U}\,\boldsymbol{K}_{aa}\boldsymbol{W}^T\boldsymbol{U}^T\,\boldsymbol{I}_{\alpha}$, $\boldsymbol{U}\,\boldsymbol{W}\boldsymbol{K}_{aa}\boldsymbol{W}^T\boldsymbol{U}^T$, $\boldsymbol{U}\,\boldsymbol{K}_{vv}\boldsymbol{U}^T$, l=1,…N_1, N_1 is the dimension of the matrix $\boldsymbol{I}_{\boldsymbol{\alpha}}$.



(a) (b)

Fig. 1. (a) Covariance matrix of a noisy signal, (b) DCT of the covariance matrix

For the design of local adaptive filters in the domain of a sliding DCT the covariance matrices and power spectra of fragments of a signal are required. Since they are often unknown, in practice, these matrices can be estimated from observed signals [14].

4 Image Enhancement in the Domain of Sliding DCT

The objective of this section is to develop a technique for noise suppression and enhancement local details on the base of a sliding DCT, and to test the algorithm performance in a real image. We design local adaptive filters to enhance noisy image. Assume that a clean image signal $\{a_k\}$ is degraded by zero-mean additive noise $\{v_k\}$:

$$x_k = a_k + v_k, \tag{14}$$

where $\{x_k\}$ is a noisy observed image. This is a particular case of (8) when $\boldsymbol{W=I}$.

Let $\{ X_l^k, A_l^k, V_l^k, \overline{A}_l^k; l= 1,..., N\}$ be the DCT transform coefficients around time k of noisy signal, clean signal, noise, and filtered signal, respectively. Here $N=2N_1+1$ is the length of the DCT. Note that N_1 is an arbitrary integer value, which is determined by the minimal size of details to be preserved after filtering.

Various criteria can be exploited to design a filter for noise removal. We use the criterion of the PMSE around time k which is defined in the domain of sliding DCT. An estimate of the reconstructed image signal with respect to the PMSE can be written as

$$\overline{A}^k(l) = \begin{cases} \left(1 - \dfrac{P_{vv}^k(l)}{P_{xx}^k(l)}\right) X^k(l), & P_{xx}^k(l) > P_{vv}^k(l) \\ 0, & \text{otherwise} \end{cases}. \tag{15}$$

where $P_{xx}^k(l) \approx \left\langle \left| X^k(l) \right|^2 \right\rangle$, $P_{vv}^k(l) \approx \left\langle \left| V^k(l) \right|^2 \right\rangle$ are estimates of the power spectra of the observed signal and noise in the domain of the sliding DCT, $\langle . \rangle$ is the average operation over signal fragments, $l= 1,..., N_1$. The obtained filter can be considered as a spectral subtraction method in the domain of sliding DCT. In general, spectral

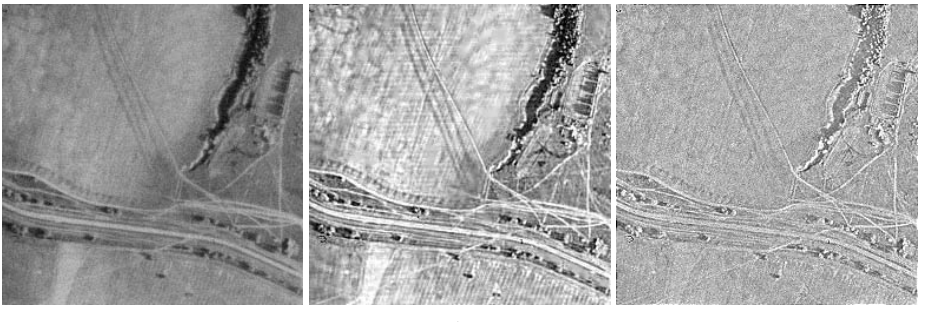
subtraction methods, while reducing the wide-band noise, introduce a new narrow-band noise due to the presence of remaining spectral peaks. To attenuate the remaining noise, one can suggest over subtraction of the power spectrum of noise by introducing a nonzero power spectrum bias.

Image enhancement may be regarded as an extension of image restoration methods. However, in contrast to image restoration, image enhancement often requires intentional distorting image signal such as increasing local contrast. In order to carry out image enhancement, we use a modification of local spectra in the domain of the sliding DCT. Finally, the processed image signal can be written as follows:

$$\overline{A}^k(l) = \begin{cases} \left(1 - \dfrac{P_{vv}^k(l)}{P_{xx}^k(l)}\right)^T X^k(l), & P_{xx}^k(l) > P_{vv}^k(l) + B^k \\ 0, & \text{otherwise} \end{cases}, \tag{16}$$

where B^k is a signal-dependent bias value, T is a parameter. When $0 \le T \le 1$, the modification of the local spectra redistributes of the energy of spectral coefficients in favor of low energy coefficients. Note that in real images the spectral coefficients of high frequency often possess low energy. The filtered image can be obtained with the use of the inverse DCT transform. Note that in the processing only the spectral coefficients with even indices are involved.

A real low contrast and noisy aerial image is shown in Fig. 2 (a). The size of image is 256x256, each pixel has 256 levels of quantization. The image is corrupted by zero-mean additive Gaussian noise. The objective of our computer experiment is to enhance meddle-sized details (about 15x15) in the noisy image.



(a) (b) (c)

Fig. 2. (a) Noisy aerial image, (b) denoised and enhanced image in the domain of sliding DCT, (c) denoised and enhanced image in the domain of global DCT

It is difficult to define a criterion to accurately quantify the performance of algorithms. Ramponi [4] proposed to calculate the local variance of the original and processed images over the pixels of 3x3 moving window. However, this method is appropriate for edge or fine detail enhancement in noise-free images. Since the objective of local detail enhancement is an improvement of visual appearance of

details, we use a subjective visual criterion. In our tests the window length of 15x15 pixels and T=0.25 is used. Since there exists difference in spectral distributions of the image signal and wide-band noise, the power spectrum of noise can be easily measured from the experimental covariance matrix [14]. The result of noise filtering and local contrast enhancing on the observed image with the proposed filter is shown in Fig. 2 (b). Fig 2 (c) shows the result of noise suppression and enhancement with the same algorithm designed in the domain of DCT of the entire image. We see that the proposed algorithm is capable to perform simultaneous noise suppression and local contrast enhancement, whereas the result of the global filtering is not satisfactory.

5 Conclusions

In this paper, we have presented a new technique for enhancing images degraded by additive noise. The technique utilizes the sliding DCT. In order to provide image processing in real time, a fast recursive algorithm for computing the sliding DCT was utilized. Extensive testing has shown that additive noise can be significantly reduced as well as local contrast enhanced by proper choice of algorithm parameters.

References

1. Oppenheim A.V., Shafer R.W. and Stockham Jr. T.G., Nonlinear filtering of multiplied and convolved signals, Proc. IEEE, Vol. 56, No. 8, (1968) 1264-1291.
2. Schreiber W.F., Wirephoto quality improvement by unsharp masking, Pattern recognition, Vol. 2, No.4, (1970) 117-121.
3. Hummel R.A., Image enhancement by histogram transformation, Comp. Graph. Image process., Vol. 6, (1977) 184-195.
4. Ramponi G., Strobel N., Mitra S.K., and Yu T., Nonlinear unsharp masking methods for image contrast enhancement, J. Electron. Imag., Vol. 5, (1996) 353-366.
5. Ramponi G., A cubic unsharp masking technique for contrast enhancement, Signal Processing, Vol. 67, (1998) 211-222.
6. Polesel A., Ramponi G., Mathews V.J., Image enhancement via unsharp masking, IEEE Trans. Image Process., Vol. 9, No. 3, (2000) 505-510.
7. Kober V., Mozerov M., Alvarez-Borrego J., Ovseyevich I.A., Unsharp masking using rank-order filters with spatially adaptive neighborhoods, Pattern Recognition and Image Analysis, Vol. 12, No.1, (2002) 46-56.
8. Oppenheim A.V., Shafer R.W., Discrete-time signal processing, Prentice Hall, Englewood Cliffs, NJ (1989).
9. Mallat S., A wavelet tour of signal processing, Academic Press, NY (1999).
10. Vitkus R.Y., and Yaroslavsky L.P., Recursive algorithms for local adaptive linear filtration, in: Mathematical Research., Academy Verlag, Berlin, (1987) 34-39.
11. Jain A.K. Fundamentals of digital image processing. Prentice Hall, NY (1989).
12. Kober V., Fast algorithms for the computation of sliding discrete sinusoidal transforms, IEEE Trans. on Signal Process., Vol. 52, No 6, (2004) 1704-1710.
13. Hou H.S. A fast recursive algorithm for computing the discrete cosine transform, IEEE Trans. Acoust. Speech Signal Process., Vol. 35, No. 10, (1987) 1455-1461.
14. Yaroslavsky L., Eden M. Fundamentals of digital optics. Birkhäuse, Boston (1996).

Qualitative Real-Time Range Extraction for Preplanned Scene Partitioning Using Laser Beam Coding

Didi Sazbon[1], Zeev Zalevsky[2], and Ehud Rivlin[1]

[1] Department of Computer Science, Technion – Israel Institute of Technology, Haifa, Israel
[2] School of Engineering, Bar-Ilan University, Ramat-Gan, Israel

Abstract. This paper proposes a novel technique to extract range using a phase-only filter for a laser beam. The workspace is partitioned according to *M* meaningful preplanned range segments, each representing a relevant range segment in the scene. The phase-only filter codes the laser beam into *M* different diffraction patterns, corresponding to the predetermined range of each segment. Once the scene is illuminated by the coded beam, each plane in it would irradiate in a pattern corresponding to its range from the light source. Thus, range can be extracted at acquisition time. This technique has proven to be very efficient for qualitative real-time range extraction, and is mostly appropriate to handle mobile robot applications where a scene could be partitioned into a set of meaningful ranges, such as obstacle detection and docking. The hardware consists of a laser beam, a lens, a filter, and a camera, implying a simple and cost-effective technique.

1 Introduction

Range estimation is a basic requisite in Computer Vision, and thus, laser techniques utilizing pattern light have been explored to a great extent. Pattern light is commonly used in a stereo configuration in order to facilitate the correspondence procedure, which forms the challenging part of triangulation. Usually, one camera is replaced by a device that projects pattern light (also known as 'structure light'), while the scene is grabbed by the other camera. A very popular group of techniques are known as 'coded structured light'. The coding is achieved either by projecting a single pattern or a set of patterns. The main idea is that the patterns are designed in such a way that each pixel is assigned with a codeword [1]. There is a direct mapping between the codeword of a specific pixel and its corresponding coordinates, so correspondence becomes trivial. Different types of patterns are used for the coding process, such as: black and white, gray scale, and RGB [2-7]. Coded structure light is considered one of the most reliable techniques for estimating range, but since usually a set of patterns is needed, it is not applicable to dynamic scenes. When using only one pattern, dynamic scenes might be allowed, but the results are usually of poor resolution.

Additional techniques implementing structured light to assist the correspondence procedure include sinusoidal varying intensities, stripes of different types (e.g. colored, cut), and projected grids [8-15]. These methods, although projecting only one pattern, still exploit a time consuming search procedure.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 320 – 327, 2005.

Here, pattern light is used only with one image to directly estimate range. No correspondence (triangulation) is needed, and the setup consists only of a laser beam, a lens, a single mask, and a camera. The main concept would be to partition the workspace into a set of range segments, in a way that would be meaningful for a working mobile robot. The motivation lies in the fact that in order to perform tasks such as obstacle detection or docking, it should be sufficient that the robot would be able to distinguish between a set of predefined ranges. The idea is to code a laser beam into different patterns, where each pattern corresponds to a specific range segment. Once a scene is illuminated by the coded beam, each patch in it would irradiate with the pattern that corresponds to its range from the light source. The beam coding is merely realized by one special phase-only filter, and consequently, the technique is accurate, fast (hardware solution), cost-effective, and in addition, fits dynamic scenes.

2 Qualitative Real-Time Range Extraction for Preplanned Scene Partitioning Using Laser Beam Coding

The proposed technique is based on an iterative design of a phase-only filter for a laser beam. The relevant range is divided into *M* meaningful planes. Each plane, once illuminated by a laser beam that propagates through the phase-only filter, would irradiate in a different, predetermined, pattern. The pattern that was chosen here consists of gratings in *M* different angles (slits). Each range would be assigned with slits having a unique angle. Once a plane is illuminated, it would irradiate with the angular slits pattern that is proportional to its range.

The iterative procedure is based on the Gerchberg-Saxton (GS) algorithm [16] as schematically illustrated in Figure 1. What follows is a description of the general concept of the algorithm. Assume we have a function denoted by $f(x, y)$, then, $f(x, y)$ could be represented as:

$$f(x, y) = |f(x, y)| \cdot \exp\{i \cdot \phi(x, y)\} \tag{1}$$

where, $|f(x, y)|$ is the amplitude of $f(x, y)$, and $\phi(x, y)$ is the phase of $f(x, y)$. We would denote the Fourier Transform of $f(x, y)$ by $F(u, v)$, thus:

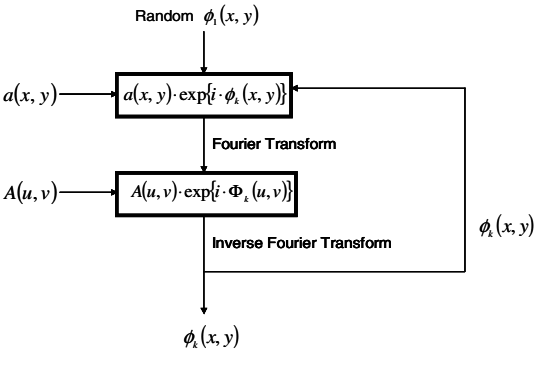


Fig. 1. A schematic description of the GS algorithm to obtain phase information

$$F(u,v)=|F(u,v)|\cdot\exp\{i\cdot\Phi(u,v)\} \tag{2}$$

where, $|F(u,v)|$ is the amplitude of $F(u,v)$, and $\Phi(u,v)$ is the phase of $F(u,v)$. Assume $|f(x,y)|$ and $|F(u,v)|$ are determined in advance and are denoted by $a(x,y)$ and $A(u,v)$, accordingly. In order to retrieve the phase, $\phi(x,y)$, from $f(x,y)$, we start with a random estimation of $\phi(x,y)$, denoted by $\phi_1(x,y)$. Thus, $f(x,y)$ is estimated by: $a(x,y)\cdot\exp\{i\cdot\phi_1(x,y)\}$. In order to design a phase-only filter, such that, using a laser beam would result in a predefined pattern, we would use $a(x,y)=1$, and $A(u,v)$ will be a function that depicts the desired pattern in which the beam, while propagating in free space, would illuminate. Although not proven mathematically, the algorithm is known to give excellent practical results.

Here, we would like to use that concept, but to create a phase-only filter that would illuminate in a pattern, that slightly changes, as a function of range. Thus, the GS algorithm should be modified to comply with these changes. The modified procedure is as follows: let $a(x,y)=1$, let $Z^j(u,v)$ be the pattern assigned to the *j*-th plane ($j=1,2,...,M$), and start with $\phi_1(x,y)$, a random estimation of $\phi(x,y)$. Proceed with the iterative procedure that is depicted schematically in Figure 2.

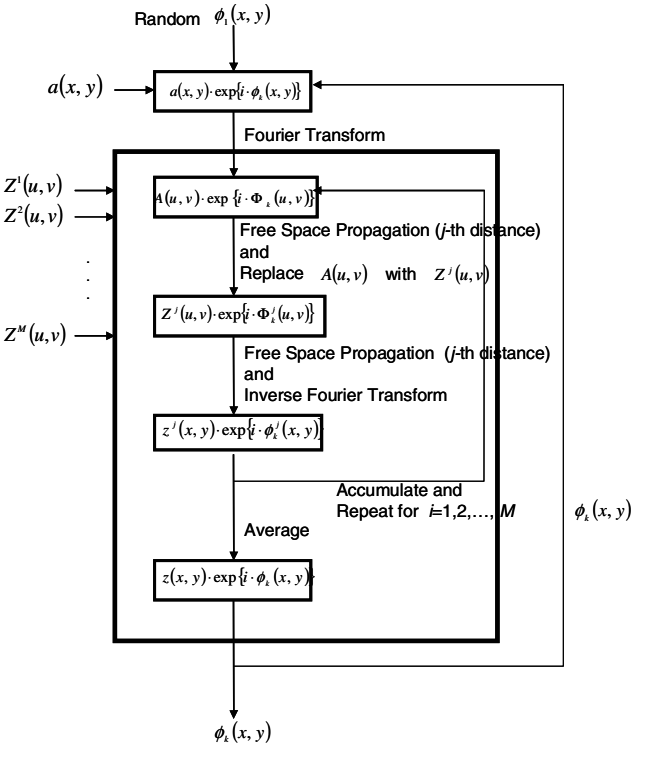


Fig. 2. A schematic description of the algorithm presented here

Note the term *Free Space Propagation* used along the procedure (Figure 2). The laser beam is propagated to the position of the plane $Z^j(u,v)$ by multiplying its spatial spectrum using the term:

$$FS(x, y, d_j) = \exp\left[\frac{2\pi i \cdot d_j}{\lambda}\sqrt{1-\left(\lambda\frac{x}{D}\right)^2-\left(\lambda\frac{y}{D}\right)^2}\right] \quad (3)$$

where, d_j is the range of the plane from the origin, λ is the wave length, and $D \times D$ are the dimensions (in meters) of the detected plane. Note also, that the planes parameters (i.e. the number of planes, the size of a plane, the location of a plane, the distances between planes – that could vary, and the patterns to be displayed on the planes) can be defined to meet specific requirements of the phase-only filter.

The expected behavior of the laser beam once illuminated and according to the physical characteristics of it would be as follows. The beam would be homogeneous until it propagates and encounters the first predefined plane, then it would exhibit the first designed slit pattern. It would keep the same pattern while propagating along the first segment until encountering the second predefined plane, then it would exhibit the second designed slit pattern. It would keep the same pattern while propagating along the second segment and so on. When it would meet the last predefined plane, it would keep propagating indefinitely with its corresponding slit pattern.

Note that the range segments can differ in length and the partitioning should not necessarily be uniform. For example, a docking mobile robot would like to decelerate first at 30 meters from the target, then at 5 meters, and again at 1 meter. The resultant phase-only filter would consist of 3 slits patterns, corresponding to the range segments of 1, 5, and 30 meters. Thus, each filter should be designed with range segments that meet the needs of the relevant task, the specific working robot, and the particular workspace.

3 Results

The proposed technique was tested with a phase-only filter designed to exhibit the patterns depicted by Figure 3, on six equally spaced planes positioned between 0.5 to 1 meters from the light source. The range between two consecutive planes equals to 0.1 meters. A laser beam having a wave length of $0.5 \cdot 10^{-6}$ meters (green light) was used. The physical size of the filter is 4×4 millimeters, and the beam was scattered in order to cover the whole filter. By using the technique described in Section 2, the

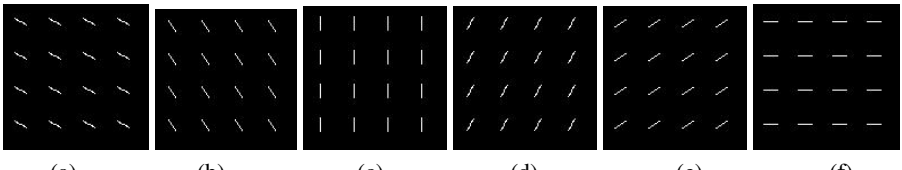


Fig. 3. The slits patterns used here

resulted filter has values in the range $[-\pi,\pi]$. In order to save in production costs, the filter was quantized into two levels: $-\pi$ and 0. As can be seen throughout the experiments, the results are satisfying, while the production is extremely cost effective.

Figure 4 shows images depicting the patterns irradiated by the phase-only filter on planes positioned at ranges 0.5, 0.6, 0.7, 0.8, 0.9, and 1 meter from the light source. Figure 5 shows the neighborhood of the patterns and the slits directions are clearly visible to the human eye. In order to automate the application and deduce the range from the assignment of a specific direction to a particular image a simple procedure can be invoked. Since these images were taken from a non calibrated camera they need simple preprocessing.

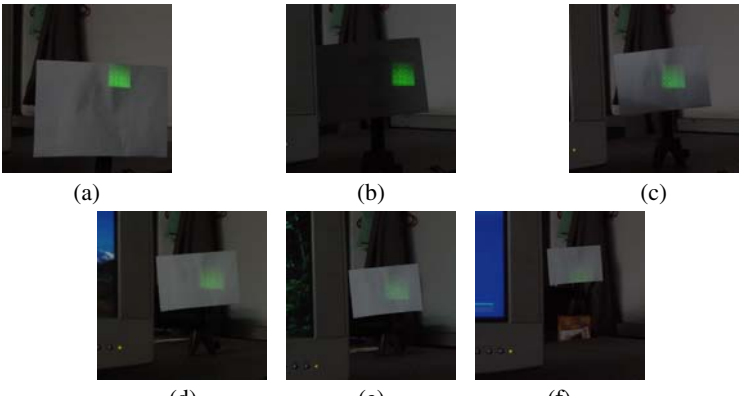


Fig. 4. The patterns irradiated by using the phase only filter at different ranges

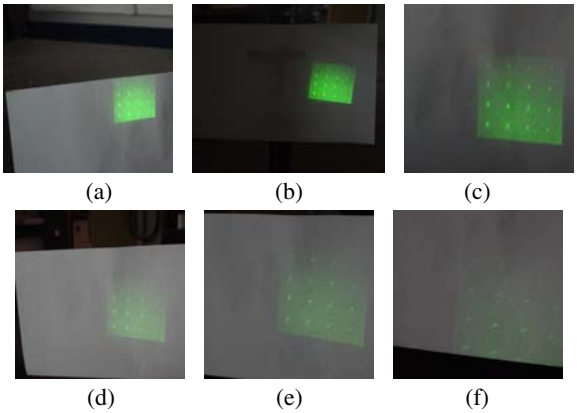


Fig. 5. The patterns of Figure 4 in closer look depicting better the slits directions

The first step consists of rotating the images, so their patterns would be aligned horizontally, and normalizing their color, so they would be comparable with the correlation patterns. The correlation patterns are merely images of the six possible slits.

Then in the next step, taking into considerations the fact that the laser is of bright green color that pops up in the images, a simple threshold is applied leaving only the relevant information. The third step is to correlate the image with the six possible patterns (i.e. slits in different directions) to get maximum response on the most compatible one. Applying this procedure confirmed that maximum correlation values corresponded to the expected patterns.

The results clearly demonstrate that using the proposed technique, range is determined immediately, in real time. In addition to its accuracy, simplicity, and speed, the technique is extremely cost effective, it comprises only of a laser beam, a lens, a filter, and a common camera.

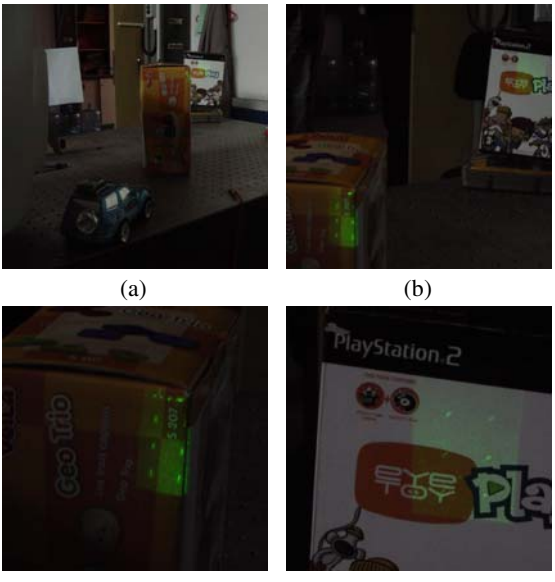


Fig. 6. A semi-realistic scene

The accuracy of the technique was measured at the border planes where the patterns were supposed to change, and found out that all the ranges were accurate up to 1-3 millimeters independently from the range itself. This implies that this specific filter has a reliability of 97% in finding the border ranges. Also, if a robot would be interested in finding out when it is exactly on a border plane, it only needs to find a point of change between two successive patterns. Considering the directions of the two patterns, the range is directly deduced. Note that, in general, if a different filter would be designed its accuracy and reliability should be measured individually.

Figure 6a depicts a semi-realistic scene where a mobile robot (the car) illuminates the scene using the proposed phase-only filter in order to detect obstacles. Two boxes, acting as obstacles, are positioned in front of it, and it can be clearly seen that the pattern of the filter is split between both of them, where one half irradiates in a specific pattern and the other half irradiates in a different pattern. For a better observation, Figure 6b consists of a closer image of the obstacles, while Figure 6c and 6d

consist of even closer images of each of the obstacles. By analyzing the patterns it can be deduced that the first obstacle is located at distance between 0.7 and 0.8 meters and the second at distance between 0.8 and 0.9 meters (in respect to the light source).

4 Discussion

A technique to qualitative real-time range estimation for preplanned scene partitioning is presented here. The setup consists of a laser beam, a lens, a single phase-only filter, and a camera. The phase-only filter is designed in such a way, that a scene patch illuminated by it, would irradiate in a unique pattern proportional to its range from the light source. The phase-only filter can be designed to meet the specific parameters of its working environment. Relevant parameter include: the location (range) of the first range segment, the number of range segments, the length of each segment (e.g. shorter for nearby environment), the uniformity of the gaps (e.g. equal, changing), the dimensions of the projected pattern (e.g. 10 centimeters, 1/2 meter), and the density of the slits forming the pattern. If the environmental conditions require a stronger contrast, a stronger laser source can be used. Note, since the physics of propagating light should be taken into considerations, the dimensions of the projected pattern are getting bigger as the range increase.

The specific scanner implemented here and described in the Results Section, is in fact, a very simple one. It could be assembled using available laboratory components. Thus, its main role in proving the correctness of the technique, and as such it was designed having a relatively short total range (0.5-1 meter) with relatively long range segments (0.1 meter), best suitable for the task of obstacle detection or docking. The environmental factors that might affect the accuracy or the reliability of this scanner are light conditioning or green obstacles. If the light is too strong, the green slits can be hardly seen. Also, if the scene would consist of green obstacles, it might be difficult to separate the slits from the background. This problem, when appropriate, can be resolved by using a laser beam of red light.

In general, the technique would mostly fit in a context of a mobile robot that would be interested in a rough estimation of a scene structure. This would enable it to identify guidelines in predetermined ranges and consequently, plan its path. The workspace can be partitioned in advance into a set of relevant ranges composed of near, intermediate, and far at the same time, with variable length of segments. Near ranges would naturally be densely segmented, whereas far ranges would be segmented in sparse manner. The robot would have its range partitioned into an appropriate and meaningful warning zones, so when a match is achieved, a corresponding action could be invoked. The technique extremely fits such scenarios by providing both qualitative and reliable results.

References

1. Salvi, J., Pages, J., Batlle, J.: Pattern Codification Strategies in Structured Light Systems. Pattern Recognition Vol. 37. (2004) 827-849
2. Caspi, D., Kiryati, N., Shamir, J.: Range Imaging with Adaptive Color Structured Light. IEEE Transactions on PAMI Vol. 20, No.5. (1998) 470-480

3. Horn, E., and Kiryati, N.: Toward Optimal Structured Light Patterns. Image and Vision Computing Vol. 17, No. 2. (1999) 87-97
4. Manabe, Y., Parkkinen, J., Jaaskelainen, T., Chihara, K.: Three Dimensional Measurement Using Color Structured Patterns and Imaging Spectrograph. ICPR'02 Vol. 3. (2002) 649-652
5. Pages, J., Salvi, J., Matabosch, C.: Implementation of a Robust Coded Structured Light Technique for Dynamic 3D Measurements. ICIP'03 Vol. 3. (2003) 1073-1076
6. Sato, K., Inokuchi, S.: Range-Imaging System Utilizing Nematic Liquid Crystal Mask. ICCV'87. (1987) 657-661
7. Valkenburg, R.J., McIvor, A.M.: Accurate 3D Measurement Using a Structured Light System. Image Vision and Computing Vol. 16. (1998) 99-110
8. Albamont, J., Goshtasby, A.: A Range Scanner with a Virtual Laser. Image and Vision Computing Vol. 21. (2003) 271-284
9. Fofi, D., Salvi, J., Mouaddib, E.M.: Uncalibrated Reconstruction: an Adaptation to Structured Light Vision. Pattern Recognition Vol. 36. (2003) 1631-1644
10. Furukawa, R., Kawasaki, H.: Interactive Shape Acquisition Using Marker Attached Laser Projector. 3DIM'03. (2003) 491-498
11. Guisser, L., Payrissat, R., Castan, S.: PGSD: an Accurate 3D Vision System Using a Projected Grid for Surface Descriptions. Image and Vision Computing Vol. 18. (2000) 463-491
12. Je, C., Lee, S.W., Park, R.-H.: High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging. ECCV'04 Vol. 1. (2004) 95-107
13. Kang, S.B., Webb, J.A., Zitnick, C., Kanade, T.: A Multibaseline Stereo System with Active Illumination and Real-Time Image Acquisition. ICCV'95. (1995) 88-93
14. Maruyama, M., Abe, S.: Range Sensing by Projecting Multiple Slits with Random Cuts. IEEE Transactions on PAMI Vol. 15, No. 6. (1993) 647-651
15. Scharstein, D., Szeliski, R.: High-Accuracy Stereo Depth Maps Using Structured Light. CVPR'03 Vol. 1. (2003) 195-202
16. Gerchberg, R.W., Saxton, W.O.: A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures. Optik Vol. 35. (1972) 237-246

A Novel Segmentation Strategy Based on Colour Channels Coupling*

Alberto Ortiz and Gabriel Oliver

Department of Mathematics and Computer Science,
University of the Balearic Islands, Spain
{alberto.ortiz, goliver}@uib.es

Abstract. A segmentation method based on a physics-based model of image formation is presented in this paper. This model predicts that, in image areas of uniform reflectance, colour channels keep coupled in the sense that they are not free to take any intensity value, but they depend on the values taken by other colour channels. This paper first enumerates and analyzes a set of properties in which this coupling materializes. Next, a segmentation strategy named C^3S and based on looking for violations of the coupling properties is proposed. Segmentation results for synthetic and real images are presented at the end of the paper.

1 Introduction

When shading is hardly noticeable in the image, as well as specularities and other optical phenomena such as inter-reflections, areas of the scene of uniform reflectance appear as regions of more or less constant colour if shadows are avoided. Nevertheless, when curved objects are imaged, the scene curvature and the objects glossiness, among others, give rise to noticeable changes in image intensity not necessarily related to object boundaries. In order to cope with these intensity variations in a more suitable way, physics-based segmentation methods embed into the logic of the algorithm a physics-based model of image formation. Among the several physics-based segmentation strategies which have been proposed so far, some of them are based on estimating directly the reflectance of the surfaces present in the scene (see for instance [1]), others look for certain configurations of clusters in colour space, as it is predicted by the Dichromatic Reflection Model proposed by Shafer [2] (by way of example, see [3,4,5]), and, finally, others use photometric invariants in their different forms ([6,7], among others).

C^3S, the segmentation method proposed in this paper, is also based on the Dichromatic Reflection Model, but it does not use any of the approaches mentioned above. Its main point, which is the most important contribution of the paper, is the use of the fact that, in uniform reflectance areas, colour channels are coupled by the reflectance of the surface material, while, in reflectance transition zones, such coupling can be broken in a number of ways. Consequently, material changes can be found by looking for violations of this coupling, which allows computing an edge map from which a segmentation of the image can be obtained. The work presented in this paper is the continuation of a research line which started in [8].

* Partially supported by project CICYT-DPI2001-2311-C03-02 and FEDER funds.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 328–335, 2005.

The rest of the paper is organized as follows: section 2 presents the image formation model considered in this work, and comments on the properties of uniform reflectance areas according to that model; sections 3 and 4 describe C^3S; section 5 presents some segmentation results for synthetic and real images; and, finally, conclusions appear in section 6.

2 Image Formation Model

2.1 General Description

It is generally accepted that objects reflection is an additive composition of body or diffuse reflection and interface or specular reflection [2]. Besides, this model is enhanced by a further term accounting for non-directional or ambient lighting which interacts with the scene increasing objects radiance irrespectively of local surface geometry. All in all, radiance at a scene point p and for a wavelength λ can be summarized as indicated in equation 1:

$$L(p;\lambda) = \overbrace{L_a(\lambda)\rho_a(p;\lambda)}^{L_a(p;\lambda)} + \overbrace{m_b(p)\left[L_d(\lambda)\rho_b(p;\lambda)\right]}^{L_b(p;\lambda)} + \overbrace{m_i(p)\left[L_d(\lambda)\rho_i(p;\lambda)\right]}^{L_i(p;\lambda)}, \quad (1)$$

where: (i) $L_a(\lambda)$ represents light coming from all directions in equal amounts while $L_d(\lambda)$ represents directional lighting; (ii) ρ_a, ρ_b and ρ_i are surface material reflectances expressing the fraction of the incoming light which is conveyed by the corresponding reflection component (ambient, body and interface, respectively), being ρ_a assumed a linear combination of the body and interface reflectances, ρ_b and ρ_i; (iii) m_b and m_i are terms dependent on local surface geometry such that $m_b, m_i \in [0, 1]$. After the process of photoelectrons integration performed in the sensor, the intensity at colour channel k can be approximated by $I^k(i,j) = C_a^k(i,j) + m_b(i,j)C_b^k(i,j) + m_i(i,j)C_i^k(i,j)$ where the composite reflectances C_a^k, C_b^k and C_i^k represent the joint contributions of lighting and the different material reflectances to the corresponding colour component.

2.2 Properties of Uniform Reflectance Image Areas

Within an image area of pixels corresponding to the same scene material, the ambient, body and interface composite reflectances are constant if the light distribution is approximately uniform throughout the area. In a noiseless environment, colour changes between image locations are, thus, only due to changes in the geometrical factors m_b and m_i, common to all the colour channels. The study of the image formation model has allowed us to express this coupling in the form of the following properties:

Property P1. *Colour channels do not cross one another.*
Property P2. *Colour channels vary in a coordinated way: when one changes, so do the others, and in the same sense, all increase or all decrease.*
Property P3. *As the intensity in one colour channel decreases, so does the difference between colour channel intensities; the opposite happens when the intensity in one channel increases.*

Table 1. Dependence of the fulfillment of properties P1-3 on the particular instantiation of the image formation model. **TP** and **AL** stand for *type of pixel* and *ambient lighting*, respectively. The NIR model assumes that $\rho_i(\lambda) = \rho_i, \forall\lambda$.

TP	**AL**	*properties fulfilled*
matte	no	P1-3 always
matte	yes	P2 always; P1 and P3 if $L_a(\lambda) = kL_d(\lambda), k > 0$
glossy	no	P1 under the NIR model and white-balanced images
glossy	yes	P1 under the NIR model, white-balanced images and $L_a(\lambda) = kL_d(\lambda), k > 0$

As it is indicated in table 1, these properties hold always in case the illumination is purely directional and the pixels considered do not show specular reflection. In the rest of cases, their general fulfillment depends on the satisfaction of the indicated conditions (see [9] for the formal proofs). However, most times all three properties hold within areas of uniform reflectance, since their unfulfillment takes place only for particular combinations of material reflectances and precise values of m_b and m_i.

Essentially, properties P1-3 lead to a set of necessary compatibility conditions between pixels, in the sense that if two pixels fail to satisfy any of the three properties for any of the two-by-two combinations between colour channels, then both pixels cannot correspond to the same scene material. Given a certain combination of colour channels, say C_1 and C_2, these three conditions can be stated geometrically, considering the space of intensity values taken by both channels, as it is shown in figures 1(a)-(c), for, respectively, properties P1-3. In all the figures and for case $I_a^1 \geq I_a^2$, the shaded areas represent the set of points (I_b^1, I_b^2) which are compatible with a certain point (I_a^1, I_a^2), in the sense of the satisfaction of the coupling properties; in case $I_a^1 \leq I_a^2$, the compatibility zone would be the complement of the shaded areas of figures 1(a) and (c) for properties P1 and P3. Therefore, if for a given (I_a^1, I_a^2), (I_b^1, I_b^2) is not inside the compatibility zone, then the corresponding pixels cannot correspond to the same scene material. Finally, figure 1(d) shows the compatibility area for all three properties simultaneously (triangles correspond to the case $I_a^1 \leq I_a^2$, while circles are for case $I_a^1 \geq I_a^2$). In view of these graphs, a compatibility relation $\mathcal{C}$ can be summarized as: (I_b^1, I_b^2) *is* $\mathcal{C}$*-compatible with* (I_a^1, I_a^2) *if: (i)* $I_b^1 \geq I_b^2$ *and the orientation of the vector joining both lies within* $[0°, 45°]$ *when* $I_a^1 \geq I_a^2$*; or (ii)* $I_b^1 \leq I_b^2$ *and the orientation of the vector joining both lies within* $[45°, 90°]$ *when* $I_a^1 \leq I_a^2$.

3 Edge Map Computation

General Discussion. Checking, at every pixel, all three properties allows computing an edge map corresponding to reflectance transitions which can be found in the image. To this end, a pixel (i_a, j_a) is considered an edge if for any of its 8 neighbours (i_b, j_b) and at least one combination of colour channels, both pixels are not compatible with one another in the sense of the relation $\mathcal{C}$ formulated in section 2.2.

Although the checking of relation $\mathcal{C}$ allows covering a considerable amount of reflectance transitions, there exist others which make manifest in such a way that the

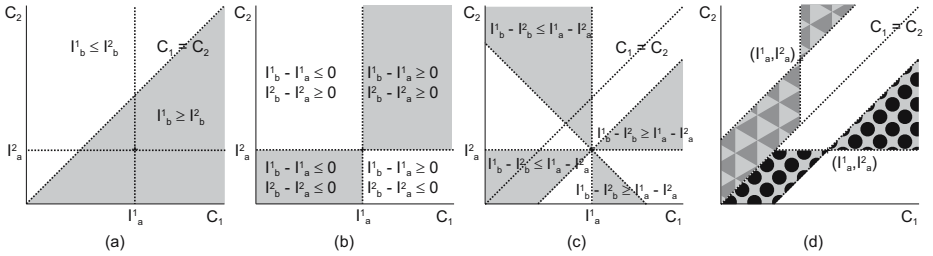


Fig. 1. Geometrical interpretation of properties P1-3

variation in the intensity does not violate any of properties P1-3 despite a reflectance transition is arising. On those cases, there is no way to decide whether the corresponding intensity changes in the colour channels are due to a reflectance transition or merely a change in m_b and/or m_i, and, thus, an ambiguity results. In other words, the interrelation between colour channels is not powerful enough on those cases so as to disambiguate the source of the intensity variation. Accordingly, knowledge from the application domain level or data from another sensor, if available, should be used to make the decision. For instance, if objects with smooth surfaces are expected in the scene, gradient information can result practical to locate this sort of reflectance transitions. In particular, LOG zero-crossings have become particularly interesting during the experiments performed. This is because LOG zero-crossings correspond to concavity-convexity changes in intensity due to the second derivative order nature of LOG, and those changes correspond quite well to the aforementioned reflectance transitions.

Implementation Issues. In order to counteract image noise in an adaptive way when computing the edge map, uncertainties related to sensor noise are used for every intensity level of every colour channel. Therefore, every digital noisy level D^k outputting the camera is associated to an interval $[D^k - \delta(D^k), D^k + \delta(D^k)]$. In this expression, the uncertainty $\delta(D^k)$ is defined as it is indicated in [9], which takes as a basis the noise model proposed in [10] and estimates the sensor performance parameters by means of a radiometric calibration of the camera [11]. With the introduction of these uncertainties, either $\mathcal{C}$ and the strategy for locating LOG zero-crossings are redefined:

- With regard to $\mathcal{C}$, given noisy intensities (D_a^1, D_a^2), a rectangular uncertainty area around (D_a^1, D_a^2) covering t uncertainties is defined, as it is depicted in figure 2(a). $\mathcal{C}$ is redefined then as: (D_b^1, D_b^2) *is* $\mathcal{C}$*-compatible with* (D_a^1, D_a^2) *if any of the points belonging to the uncertainty area of* (D_b^1, D_b^2) *falls within the union of the compatibility zones of the points belonging to the uncertainty area of* (D_a^1, D_a^2) *(shaded area of figure 2(a))*. The pixel corresponding to (D_a^1, D_a^2) is thus considered an edge only if there is no possibility of (D_b^1, D_b^2) being $\mathcal{C}$-compatible with (D_a^1, D_a^2).
- As for LOG zero-crossings, whenever a zero-crossing is detected, it is classified as relevant if the positive and negative peak LOG values along the direction of detection are larger than t times the respective uncertainties. Those uncertainties are calculated using standard uncertainty propagation rules, by which, if the output of the operator is calculated as $f = \sum_x D^k(x)m(x)$, where $m(x)$ would be the LOG

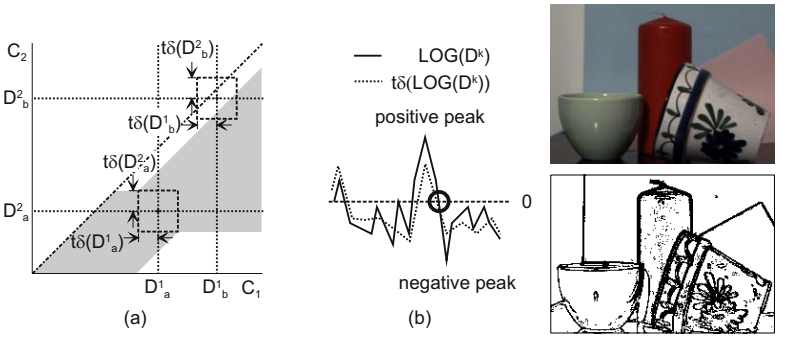


Fig. 2. Left: (a) Redefinition of $\mathcal{C}$ (the uncertainty areas have been magnified for illustration purposes); (b) Only the LOG zero-crossing behind the circle is considered relevant. Right: Edge map for image on top.

mask constants, then $\delta(f) = \sqrt{\sum_x \delta(D^k(x))^2 m^2(x)}$ [12]. By way of example, see figure 2(b), where only the zero-crossing behind the circle would be considered relevant.

To finish about the computation of the edge map: (1) although a reflectance transition should theoretically involve just two pixels, in real images they tend to span along several image cells because of real cameras' aliasing, so that the edge map must in general be expected to consist of thick edges; and (2) pixels around specularities can be labelled as edges, because, although not always edges related to the $\mathcal{C}$ relationship are found near specularities, relevant LOG zero-crossings can clearly be found, particularly for "steep" specularities. The two above-mentioned facts can be easily observed in figure 2(right).

4 Colour Channel Coupling-Based Segmentation (C^3S)

In general, segmenting an image consists in grouping pixels in homogeneous regions, generally according to a certain perceptually-based homogeneity criterion. Most times, one is interested in the image regions corresponding to the same perceptual colour. In physical terms, this means grouping pixels in uniform reflectance areas.

The experiments performed have shown that the closed regions contained in the edge map described in section 3 tend to show uniform reflectance. Therefore, a first partition of the image can be obtained if connected components not including edge pixels are found and edges are then added to the nearest most similar connected component. To perform this last task, *Principal Component Analysis* is used to characterize the cluster in colour space corresponding to every connected component. As the edge map is presumed to separate pixels corresponding to "steep" specularities from matte pixels, and according to the Dichromatic Reflection Model, the corresponding regions are expected to be describable by linear or point clusters in colour space.

In order not to mix pixels corresponding to different scene materials, connected components without a linear or point-wise shape in colour space are discarded. Those

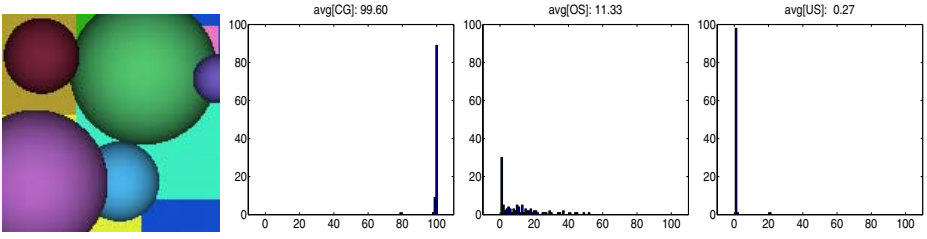


Fig. 3. (1st) example of noisy synthetic image; (2nd,3rd,4th) CG, OS and US histograms

connected components result when a contour is not finished due to noise and a path of non-edge pixels between two regions of different reflectance is created. On those cases, the corresponding pixels are grouped again controlling the growth of the corresponding cluster in colour space as the pixels are added. When the cluster dimensionality exceeds that of a line, the growing of the connected component stops and a new connected component is created and grown. The process continues until no more pixels of the original connected component are unlabelled.

A region merging stage based on a possibilistic clustering algorithm [13] follows next, in order to remove the probably low, but not generally zero, degree of oversegmentation which can result.

5 Experimental Results and Discussion

Several results for synthetic and real curvature-dominated images are given below to prove experimentally the usefulness of C^3S.

On the one hand, a set of 100 synthetic colour images of different scenes consisting of spheres and planes were generated. Besides, noise was incorporated conforming with the camera model described in [11], parameterized according to the performance of the camera used in the experiments with real images. All 100 images were then segmented and the percentage of correctly grouped pixels (CG), oversegmentation (OS) and undersegmentation (US) were determined for every image (CG corresponds to pixels belonging to regions which are not spread among several regions of the reference segmentation, while OS accounts for pixels belonging to regions splitting an only true region; US, finally, covers pixels in regions spanning several true regions). Figure 3 shows an example of noisy synthetic image, together with histograms for CG, OS and US. As a global result of the experiment: $\overline{\mathrm{CG}} = 99.60\%$, $\overline{\mathrm{OS}} = 11.33\%$ and $\overline{\mathrm{US}} = 0.27\%$.

On the other hand, a set of real images, standard and non-standard, were also considered. For the non-standard images, the gamma correction switch of the camera was set to 1 in order to get images whose intensity is proportional to the light striking the sensor, which is essential for physics-based vision algorithms. Although standard images do not typically satisfy this constraint, they were also considered in the experiments for comparison with other segmentation algorithms.

Figure 4 shows two non-standard and two standard images, with the region contours produced by C^3S overimposed over the original images. Besides, for comparison purposes between a physics-based and a non-physics-based approach, region contours

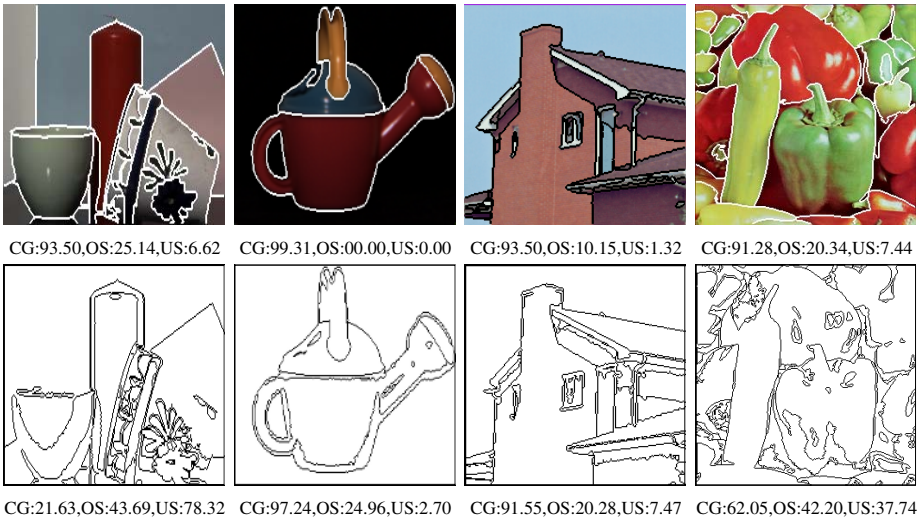


Fig. 4. (first row) C^3S; (second row) C&M

Fig. 5. More results for C^3S and real images

resulting for the recognized mean-shift based segmentation algorithm by Comaniciu and Meer [14] (C&M) are also given below each image (the undersegmentation option was used in all cases). Values for CG, OS and US are also provided for both sets of segmentation results, having used, on those cases, reference segmentations manually determined. Figure 5 shows more results for non-standard images.

As for parameters, the standard deviation for LOG was set to 1.0 in all cases. The number of uncertainties considered for detecting edges related with relation $\mathcal{C}$, t_1, and the number of uncertainties for capturing LOG zero-crossings, t_2, were set to 3 in the experiment with synthetic images, while t_2 needed a finer tuning when segmenting real images. Nevertheless, a restricted set of values (1.5 or 2) was necessary to achieve good segmentations.

As can be observed, C^3S is able to deal correctly with scene curvature, while C&M tends to produce oversegmentation on those cases. Furthermore, in cases where C&M mixes pixels from different objects, C^3S finds the correct classification (first image of figure 4). Sometimes, however, object edges give rise to a certain degree of oversegmentation in C^3S output because of the ambiguity in the reflectance transitions detected by

means of LOG zero-crossings (first image of figure 4); a similar behaviour is observed for shadows. The third image in figure 5 is also interesting because of, contrarily to expected, the right behaviour of C^3S with the textured cup. As for the standard images, both algorithms provide a similar output in the *house* image, while in the *peppers* image, where curved objects appear, C^3S clearly outperforms C&M. (See [9] for more experimental results, intermediate and final, and a more detailed discussion, not included here due to lack of space)

6 Conclusions

C^3S, a curvature-insensitive segmentation method based on a physics-based image formation model, has been proposed. The method uses the coupling between colour channels in uniform reflectance areas to locate image locations where colour channels turn out to be uncoupled because of a reflectance transition arising there. Experiments with synthetic and real images have been presented, showing the power of the approach for dealing with scenes with curved objects and different surface materials.

References

1. Lee, C.H., Rosenfeld, A.: Albedo estimation for scene segmentation. PRL **1** (1983) 155–160
2. Shafer, S.: Using color to separate reflection components. COLOR Research and Application **10** (1985) 210–218
3. Klinker, G., et al.: A physical approach to color image understanding. IJCV **4** (1990) 7–38
4. Kroupnova, N.: Method for multi-spectral images segmentation based on the shape of the colour clusters. In: SPIE - Human Vision, Vis. Proc., and Dig. Display VI. (1996) 444–453
5. Ong, C.K., Matsuyama, T.: Robust color segmentation using the dichromatic reflection model. In: Proceedings of ICPR. (1998) 780–784
6. Gevers, T.: Adaptive image segmentation by combining photometric invariant region and edge information. PAMI **24** (2002) 848–852
7. Healey, G.: Segmenting images using normalized color. SMC **22** (1992) 64–73
8. Ortiz, A., Oliver, G.: Detection of colour channels uncoupling for curvature-insensitive segmentation. In: Proc. of the Iberian Conference on Pattern Recognition and Image Analysis. (2003) 664–672
9. Ortiz, A., Oliver, G.: Segmentation of images based on the detection of reflectance transitions. Technical Report A-3-2003, Departament de Matemàtiques i Informàtica (Universitat de les Illes Balears) (2003)
10. Healey, G., Kondepudy, R.: Radiometric CCD camera calibration and noise estimation. PAMI **16** (1994) 267–276
11. Ortiz, A., Oliver, G.: Radiometric calibration of CCD sensors: Dark current and fixed pattern noise estimation. In: Proc. of ICRA. Volume 5. (2004) 4730–4735
12. Taylor, J.: An Introduction to Error Analysis. University Science Books (1997)
13. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press (1999)
14. Comaniciu, D., Meer, P.: Robust analysis of feature spaces: Color image segmentation. In: Proc. of CVPR. (1997) (The code is publicly available in http://www.caip.rutgers.edu/riul/research/papers/abstract/feature.html).

Seeded Watersheds for Combined Segmentation and Tracking of Cells

Amalka Pinidiyaarachchi[1,2,⋆] and Carolina Wählby[1]

[1] Centre for Image Analysis, Uppsala University, Sweden
[2] Dept. of Statistics and Computer Science, University of Peradeniya, Sri Lanka
amalka@cb.uu.se, carolina@cb.uu.se

Abstract. Watersheds are very powerful for image segmentation, and seeded watersheds have shown to be useful for object detection in images of cells in vitro. This paper shows that if cells are imaged over time, segmentation results from a previous time frame can be used as seeds for watershed segmentation of the current time frame. The seeds from the previous frame are combined with morphological seeds from the current frame, and over-segmentation is reduced by rule-based merging, propagating labels from one time-frame to the next. Thus, watershed segmentation is used for segmentation as well as tracking of cells over time. The described algorithm was tested on neural stem/progenitor cells imaged using time-lapse microscopy. Tracking results agreed to 71% to manual tracking results. The results were also compared to tracking based on solving the assignment problem using a modified version of the auction algorithm.

1 Introduction

Cell migration analysis is of interest in many different biological applications, e.g. when studying leucocytes, fibroblasts, or cancer cells [1,2,3]. This paper is part of a greater project with the aim to aid in the investigation of the mechanisms behind differentiation of neural stem/progenitor cells by creating a system for analyzing the motion of such cells in vitro, previously described in [4]. Once the cells have been imaged using time-lapse microscopy, the individual cells in each image have to be automatically segmented and tracked. Cells in tissue culture are non-rigid, irregularly shaped bodies. As the cells move, grow and divide they take on a variety of different shapes and the contrast between the cell boundary and the image background varies continually. Several techniques for cell image segmentation have been proposed in the recent years. A simple and popular method is thresholding, based on histogram characteristics of the pixel/voxel intensities of the image [5]. However, transition between object and background may be diffuse, making an optimal threshold level, and thereby an accurate description of the cellular shape, difficult to find. In addition, thresholding techniques do not guarantee closed object contours, and often require

⋆ The first author was funded by the SIDA-IT project.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 336–343, 2005.

substantial post-processing [10,11]. Instead of defining the border between object and background by a threshold, similarity in intensity, gradient, or variance of neighboring pixels can be used. This is usually referred to as region growing, and includes techniques such as watershed segmentation [12,13]. Watershed segmentation often results in over-segmentation due to intensity variations within both objects and background. Seeded watersheds, where the number of segmentation boundaries is limited to a set of input seeding regions, have been used for segmentation of cell nuclei in images produced by fluorescence microscopy [14]. Fully automatic seeding often results in more than one seed per object, or objects containing no seed at all. Over-segmentation caused by extensive seeding can however be reduced by model-based region merging [15].

To improve the segmentation, the segmentation and tracking can be combined. A popular method that combines the segmentation and tracking is active contours (snakes), previously used for cell segmentation/tracking in e.g. [1,16,17]. A significant drawback with the traditional snake is its inability to deal with splitting cells. Another technique combining segmentation and tracking is demonstrated in [2], where the directional movement of cells induced by a direct current (galvanotaxis) is studied. Standard thresholding combined with clustering results in a coarse segmentation, which is refined further using the result of an association/tracking algorithm. Tracking is done using a modified version of the Kalman filter. This method works well because of the known directional movement of the cells. However, assumptions about a cells state equations are potentially risky in cases where little is known about the laws governing the cell motion, and when the purpose of the analysis is to gain more information about cell kinematics. Other common tracking techniques are the nearest-neighbor method [18] and correlation matching [6]. Nearest-neighbor techniques require low cell density and high temporal sampling rate, while correlation matching instead requires the individual cells features, such as size, shape and intensity, to be fairly constant between two consecutive frames, which is not often the case for cells.

This paper presents a combined segmentation and tracking method based on seeded watersheds. The presented technique is compared to manual tracking results as well as with a previous approach using a combination of nearest-neighbor and correlation matching techniques where the cells are associated using the auction algorithm [4,7].

2 The Imaging System

The time-lapse microscopy and imaging system has previously been described in [4], and will only be discussed briefly. For cell tracking experiments, cells were cultured in a microscope incubator and imaged every 10 minutes for 45.5 hours. Time-lapse microscopy of neural stem/progenitor cells in vitro requires a sterile environment and a feedback control of temperature and pH. This was achieved using a closed incubation chamber, a heater unit and warm, filtered air together with a CO_2 mixture (regulating the pH) circulating within the acrylic chamber, all regulated by a controller unit. An inverted light microscope with

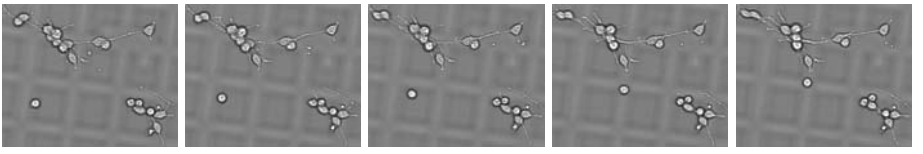


Fig. 1. Cut-outs from five consecutive time frames, each 10 minutes apart. As can be seen, some cells are fairly stationary, while others move faster.

a 20X ordinary bright field objective was used for imaging. No special contrast method such as phase contrast, or DIC, was used. By slight defocusing, higher contrast is traded for lower resolution. Images were captured by a CCD video camera and the specimen was moved in the horizontal plane by a motorized microscope stage to image the separate wells. A focus stack was captured by sweeping the approximate interval, and the image with the best contrast was selected by an auto-focus function searching for the optimal focus position based on local image contrast. A slight image drift in both x and y direction between consecutive image frames was corrected by estimation of the translation for each pair of frames using the correlation in the Fourier domain. The images used in this paper were 30 different image sequences acquired at two different occasions. Each image sequence contained 273 gray scale frames, each of size 634 by 504 pixels. A series of cut-outs from five consecutive time frames is shown in Fig. 1.

3 Segmentation and Tracking

Cell segmentation and tracking over time is performed as a combined procedure.

Initial Seeding. The image segmentation is based on seeded watershed segmentation. First, seeds representing objects (cells) and background are defined. As the image intensities of the cells show a range of values partly overlapping with the background intensities, see Fig. 2(a), intensity thresholding will not separate cells from background in a satisfactory way. A more effective method for separating objects from background is using variance measures[10]. A 10 by 10 variance filter is applied to the image, and the image background seed is defined by a fixed intensity threshold (t=300) in the variance image, shown in Fig. 2(b). The resulting background seed is shown in gray in Fig. 2(c). Object seeds at the initial step are found using the extended h-maxima transform [8] that filters out local maxima using a contrast criterion. All maxima with heights smaller than a threshold level h are suppressed. A low h will result in many seeds, often more than one seed per cell, while a high h will leave some cells without a seed. Since the subsequent merging step (described below) reduces over-segmentation due to extensive seeding, a comparably low h value (h=16) produces a suitable seed. All foreground seeds are uniquely labeled using connected component labeling. The seeds obtained from the h-maxima transform (referred to as h-maxima seeds hereafter) are then combined with the background seed identified by the variance

thresholding. The background seed is assigned the highest label after labeling of object seeds. Object (white) and background seeds (gray) are shown in Fig. 2(c).

Watershed segmentation. The combined object and background seed information is used as input for watershed segmentation. Seeded watershed segmentation is applied to the inverse of the original intensity image. The dark edges of the cells are thus interpreted as ridges, and the brighter cells and background as shallow valleys in the watershed landscape. Seeds represent holes in the landscape, and as the landscape is submerged in water, the water will start to flow into the minima of the landscape, creating one catchment basin associated with each local minima. As the water rises, water from neighboring catchment basins will meet. At every point where two catchment basins meet, a dam, or watershed is built. These watersheds are the segmentation of the image. The speed of watershed segmentation is improved using sorted pixel lists [9]. The initial result of seeded watershed segmentation of Fig. 2(a) using the seeds from Fig. 2(c) is shown in Fig. 2(d).

Merging. Watershed segmentation often results in over-segmentation. The segmentation is therefore followed by two merging steps. The first merging step removes extra regions due to non-seeded local minima. These regions are merged with the neighboring region towards which it has its weakest border. The weakest border is defined as the border in which the mean intensity of the inverse of the original image is the least [15]. The seeded neighbor may be either object or background, and the merging continues until all non-seeded objects are merged. See the result of the first merging step in Fig. 2(e). The second merging step deals with over-segmentation resulting from extensive h-maxima seeding. This over-segmentation is reduced by removing region boundaries crossing bright parts of the image, e.g., a boundary dividing a bright cell in two. In this case we continue the merging until all remaining objects have a defined maximum average intensity along the border (t=100). This step will not only reduce over-segmentation, but also merge false objects, such as debris, with the background. The final segmentation result is shown in Fig 2(f).

Tracking by propagation of seeds. The resulting segmentation is then processed to obtain seeds that are used for tracking the objects over time. The centroid of each region is computed and these centroids are then dilated with a 3 by 3 structuring element. These seeds (referred to as centroid seeds hereafter) are labeled according to the preceding segmentation. Each centroid seed is compared with the h-maxima seeds of the next frame to identify any overlap between the two seeds. The most frequent value of the overlap is chosen when updating the label values of the h-maxima seeded frame. H-maxima seeds that do not overlap with centroid seeds keep their original label, and centroid seeds that do not overlap with h-maxima are transferred directly to the new set of seeds. In order to make the check efficient, initial labeling of the h-maxima seeds are adjusted to take the values starting from the highest value of the labels from the previous frame. This enables an easy implementation of the checking process with the trade off of label values increasing fast if no overlap is found. The

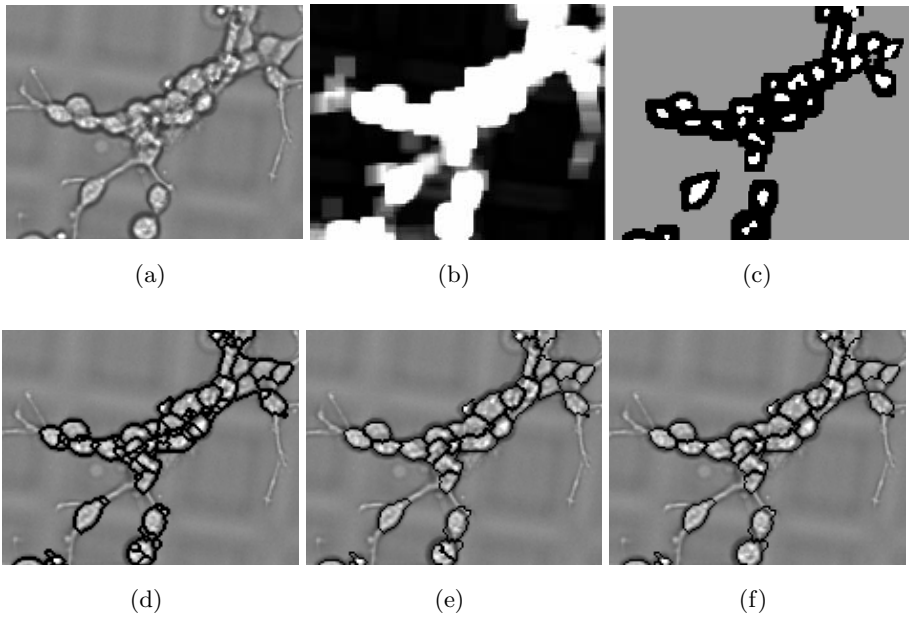


Fig. 2. The steps of the segmentation algorithm. (a) Original image. (b) Variance image. (c) Foreground seeds (white) and background seeds (gray). (d) Result of initial seeded watershed segmentation. (e) Result after merging to remove non-seeded objects. (f) Final result after second merging to remove over-segmented objects.

combined set of seeds together with the background seed is thereafter used as input for watershed segmentation followed by the merging steps described above. This yields a result where cells inherit their labels from the previous frame if a correspondence is found or are assigned a new unique label if no correspondence is found.

Implementation. The segmentation and tracking method was prototyped using Matlab (The MathWorks, Inc.) together with the Image Processing Toolbox and some functions from the DipImage toolbox (The Pattern Recognition Group of the TU, Delft).

4 Results

Validation of automatic segmentation and tracking demands image sequences of correctly tracked cells to which the automatic tracking results can be compared. Generation of ground truth for large image sequences is a very time consuming task as it has to be done by manual inspection of each image frame. A user-friendly graphical application for manual correction of segmentation and tracking results described in [4]was used and segmentation errors were detected by

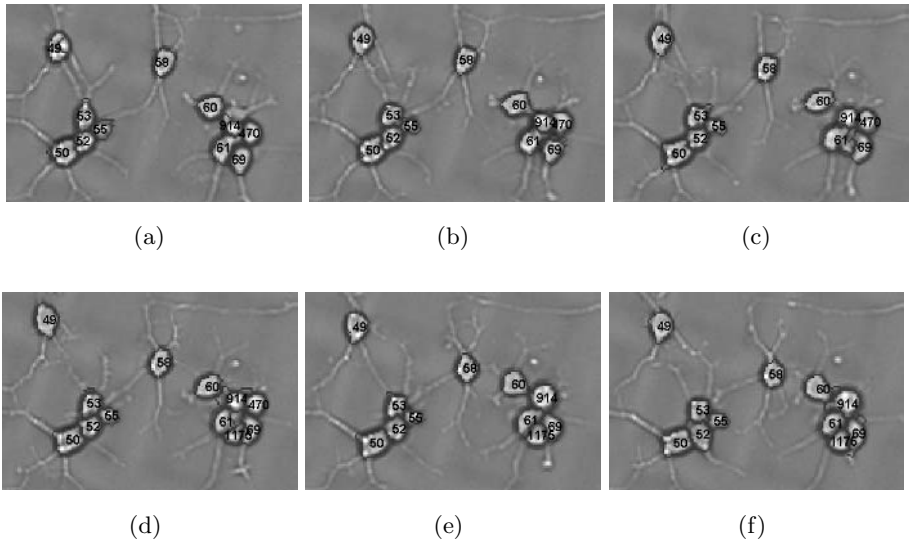


Fig. 3. The segmentation and tracking result (represented by a number printed on each cell) of 6 consecutive time frames. The numbers change when cells split (cell 61 becomes 61 and 1175 in d) and merge (914 and 470 become 914 in e).

pairwise comparison of automatically segmented and tracked frames and manually corrected ground truth. Four types of segmentation errors were identified: over-segmented cells, under-segmented cells, partially detected cells, and missed cells. Tracking errors were defined as changes of labels between consecutive image frames. If a single cell changes label once during a 100-frame sequence, it is considered as a 1% error. As the same cell can be correctly segmented but incorrectly tracked between two image frames, or incorrectly segmented but correctly tracked, we count tracking errors only in frames that are correctly segmented.

A total of 185 cells were chosen for manual tracking based on the result of an immunostaining performed after the completed time-lapse experiment [4]. Tracks represent a fairly random selection of cells available in the final image of each time-lapse experiment. Images were acquired every 10 minutes for 45.5 hours. During this comparatively long observation time, cells can move long distances, and disappear from the field of view. The cells also divide, and group in clusters where they can no longer be tracked, not even by manual inspection. Therefore, only 28% of the cells chosen for manual correction were present for the full 45.5 hour experiment. The total number of cells that were correctly segmented and tracked over time divided by the total number of tracked cells over time was 71.3%. The corresponding error rate for tracking using our previous approach based on a combination of nearest-neighbor and correlation matching techniques was 88%. The difference in corrections made by two different users correcting the same sequence, i.e. inter-observer variability, was estimated to 2.5%. This can be used as a measure of the variance of the error rate of the corrected sequences. A series of tracking results is shown in Fig. 3.

5 Discussion

This paper presents a combined segmentation and tracking method based on seeded watersheds and rule-based merging. It allows for object splitting, addition of new objects, and disappearance of objects. It does however fail if the objects move more than one object radius between consecutive frames. The method was tested the complex problem of analyzing cell migration on the basis of light microscopy image sequences. Cells sometimes congregate, forming large cell clusters. In these clusters, individual cells are difficult, or even impossible to resolve. This causes problems both for the segmentation and tracking. Unfortunately, we have found that growing the cells at a lower density is not a solution, since the cells are more likely to die if seeded to sparsely. The error counting method differs significantly from that of Zimmer et al. [16] that requires a cell to be correctly tracked from the first frame to the last, or it is counted as an error. The time interval also differs from 1 s between two frames in their method to 10 minutes in ours giving a completely different type of image sequences to work with. The advantage of this method is its simplicity in implementation and the ability to perform both segmentation and tracking together. The method could be used very effectively for tracking objects when there is less clustering and less rapid motion than in the complex set of images we have tested it on. This can also be used as the first step to obtain a result that is used as the input to a more detailed method that addresses such complexities. The fact that the simple steps used in order to segment and track the objects giving an encouraging result proves that, with further improvements it can become as effective as any other similar method.One such improvement will be to have further dilation on the centroid seeds in cases a correspondence could not be found.The current implementation takes a few hours to run on an image sequence of about two hundred cells and 250 time-frames. Further studies will be done with the intention of reducing the processing time of large sequence of images in Matlab by combining its capabilities with other environments such as C/C++.

Acknowledgments. The authors would like to thank K. Althoff, J. Faijerson, J. Degerman and T. Gustavsson for providing image data.

References

1. F. Leymarie and M. Levine: Tracking deformable objects in the plane using an active contour model, IEEE Trans. Pattern Anal. Machine Intell., **15**(6) 617-634 (1993)
2. T. Kirubarajan, Y. Bar-Shalom, and K. Pattipati: Multiassignment for tracking a large number of overlapping objects, IEEE Trans. Aerosp. Electron. Syst., **37**(1) 2-21 (2001)
3. P. Umesh Adiga and S. Chaudhuri: Segmentation of volumetric histopathological images by surface following using constrained snakes, in Proc. of 14th Int. Conf. Pattern Recogn., **2** 1674-1676 (1998)

4. K. Althoff, C. Wählby, J. Faijerson, J. Degerman, A. Pinidiyaarachchi, M. Gedda, P. Karlsson, T. Olsson, P. Eriksson, E. Bengtsson, T. Thorlin, and T. Gustavsson: Time-lapse microscopy and image analysis for in vitro cell migration analysis. (submitted)
5. P. Sahoo, S. Soltani, A. Wong, and Y. Chen: A survey of thresholding techniques, Comp. Vis. Graph. Im. Proc., **41** 233-260 (1988)
6. T. Gustavsson, K. Althoff, J. Degerman, T. Olsson, A.-C. Thoreson, T. Thorlin, and P. Eriksson: Time-lapse microscopy and image processing for stem cell research modeling cell migration, in Medical Imaging 2003: Image Processing, **5032** 1-15 (2003)
7. D. Bertsekas: Auction algorithms, in Linear Network Optimization: Algorithms and Codes, pp. 167-244, The MIT Press, 1 ed. (1991)
8. P. Soille, Morphological Image Analysis: Principles and Applications. Springer-Verlag (1999)
9. L. Vincent and P. Soille: Watersheds in digital spaces: an efficient algorithm based on immersion simulations, IEEE Trans. Pattern Anal. Machine Intell., **13**(6) 583-598 (1991)
10. K. Wu, D. Gauthier, and M. Levine: Live cell image segmentation, IEEE Trans. Biomed. Eng., **42**(1) 1-12 (1995)
11. C. Ortiz de Solorzano, E. Garcia Rodriguez, A. Jones, D. Pinkel, J. Gray, D. Sudar, and S. Lockett: Segmentation of confocal microscope images of cell nuclei in thick tissue sections, Journal of Microscopy, **193** 212-226 (1999)
12. S. Beucher and C. Lantuéjoul: Use of watersheds in contour detection, in Int. Workshop on Image Processing, CCETT, Rennes, France (1979)
13. L. Vincent: Morphological grayscale reconstruction in image analysis: applications and efficient algorithms, IEEE Trans. Image Processing, **2**(2) 176-201 (1993)
14. G. Landini and E. Othman: Estimation of tissue layer level by sequential morphological reconstruction, Journal of Microscopy, **209**(2) 118-125 (2003)
15. C. Wählby, I.-M. Sintorn, F. Erlandsson, G. Borgefors, and E. Bengtsson: Combining intensity, edge, and shape information for 2D and 3D segmentation of cell nuclei in tissue sections, Journal of Microscopy, **215**(1) 67-76 (2004)
16. C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J.-C. Olivo- Marin: Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing, IEEE Trans. Med. Imag., **21**(10) 1212-1221 (2002)
17. O. Debeir, I. Camby, R. Kiss, P. Van Ham, and C. Decaestecker: A model-based approach for automated in vitro cell tracking and chemotaxis analyses, Cytometry, **60A** 29-40 (2004)
18. Z. Demou and L. McIntire: Fully automated three-dimensional tracking of cancer cells in collagen gels: Determination of motility phenotypes at the cellular level, Cancer Research, **62** 5301-5307 (2002)

Image Segmentation Evaluation by Techniques of Comparing Clusterings

Xiaoyi Jiang[1], Cyril Marti[2], Christophe Irniger[2], and Horst Bunke[2]

[1] Department of Computer Science, University of Münster Einsteinstrasse, 62, D-48149 Münster, Germany
xjiang@math.uni-muenster.de
[2] Institute of Informatics and Applied Mathematics, University of Bern Neubrückstrasse, 10, CH-3012 Bern, Switzerland
{marti, iriniger, bunke}@iam.unibe.ch

Abstract. The task considered in this paper is performance evaluation of region segmentation algorithms in the ground truth (GT) based paradigm. Given a machine segmentation and a GT reference, performance measures are needed. We propose to consider the image segmentation problem as one of data clustering and, as a consequence, to use measures for comparing clusterings developed in statistics and machine learning. By doing so, we obtain a variety of performance measures which have not been used before in computer vision. In particular, some of these measures have the highly desired property of being a metric. Experimental results are reported on both synthetic and real data to validate the measures and compare them with others.

1 Introduction

Image segmentation and recognition are central problems of computer vision for which we do not yet have any solution approaching human level competence. Recognition is basically a classification task and one can empirically estimate the recognition performance (probability of misclassification) by counting classification errors on a test set. Today, reporting recognition performance on large data sets is a well accepted standard. In contrast, segmentation performance evaluation remains subjective. Typically, results on a few images are shown and the authors argue why they look good. The readers never know whether the results have been opportunistically selected or are typical examples, and how well the demonstrated performance extrapolates to larger sets of images.

The main challenge is that the question "To what extent is this segmentation correct" is much subtler than "Is this face from person x". While a huge number of segmentation algorithms have been reported, there is only little work on methodologies of segmentation performance evaluation [9]. Several segmentation tasks can be identified: edge detection, region segmentation, and detection of curvilinear structures. In this work we are concerned with region segmentation. In addition we follow the GT-based evaluation paradigm[1], in which some refer-

[1] Other paradigms include theoretical approaches, non-GT based and task-based techniques, see [9] for details.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 344–351, 2005.

ence segmentation result (ground truth) is available and the task is to measure the difference between the machine segmentation result and the ground truth.

We propose to consider the image segmentation problem as one of data clustering and, as a consequence, to use measures for comparing clusterings developed in statistics and the machine learning community for the purpose of segmentation evaluation. We start with a short discussion of related work. Then, measures for comparing clusterings are presented, followed by their experimental validation. Finally, some discussions conclude the paper.

2 Related Work

In [5] a machine segmentation (MS) of an image is compared to the GT specification to count instances of correct detection, under-segmentation, over-segmentation, missed regions, and noise regions. These measures are based on the degree of mutual overlap required between a region in MS and a region in GT, and are controlled by a threshold T. This evaluation method is widely used for texture segmentation [2] and range image segmentation [5,7,8,12,13].

In contrast, the approach from [6] delivers one single performance measure. For each MS region R one finds the GT region R^* with the maximum intersection. Then, the total intersection between R and all GT regions other than R^* is used to compute an overall difference measure between MS and GT.

In [10] another single overall performance measure is proposed. It is designed so that if one region segmentation is a refinement of another (at different granularities), then the measure should be small or even zero. Due to its tolerance of refinement this measure is not sensible to over- and under-segmentation and may be therefore not applicable in some evaluation situations.

3 Measures for Comparing Clusterings

Given a set of objects, $O = \{o_1, \ldots, o_n\}$, a clustering of O is a set of subsets $C = \{c_1, \ldots, c_k\}$ such that $c_i \subseteq O$, $c_i \cap c_j = \emptyset$ if $i \neq j$, $\bigcup_{i=1}^{k} c_i = O$. Each c_i is called a cluster. Clustering has been extensively studied in the statistics and machine learning community. In particular, several measures have been proposed to quantify the difference between two clusterings $C_1 = \{c_{11}, \ldots, c_{1k}\}$ and $C_2 = \{c_{21}, \ldots, c_{2l}\}$ of the same set O.

If we interpret an image as a set O of pixels and a segmentation as a clustering of O, then these measures can be applied to quantify the difference between two segmentations, e.g. between MS and GT. This view of the segmentation evaluation tasks opens the door for a variety of measures which have not been used before in computer vision. As we will see later, some of the measures are even metrics, being a highly desired property which is not fulfilled by the measures discussed in the last section. In the following we present three classes of measures.

3.1 Distance of Clusterings by Counting Pairs

Given two clusterings C_1 and C_2 of a set O of objects, we consider all pairs of objects, $(o_i, o_j), i \neq j$, from $O \times O$. A pair (o_i, o_j) falls into one of the four categories

- in the same cluster under both C_1 and C_2 (The total number of such pairs is represented by N_{11})
- in different clusters under both C_1 and C_2 (N_{00})
- in the same cluster under C_1 but not C_2 (N_{10})
- in the same cluster under C_2 but not C_1 (N_{01})

Obviously, $N_{11} + N_{00} + N_{10} + N_{01} = n(n-1)/2$ holds where n is the cardinality of O.

Several distance measures, also called indices, for comparing clusterings are based on these four counts. The Rand index, originally introduced in [14], is defined as

$$\mathcal{R}(C_1, C_2) \;=\; \frac{N_{11} + N_{00}}{n(n-1)/2}$$

Fowlkes and Mallows [4] introduce the following index

$$\mathcal{F}(C_1, C_2) \;=\; \sqrt{W_1(C_1, C_2) W_2(C_1, C_2)}$$

as the geometric mean of

$$W_1(C_1, C_2) \;=\; \frac{N_{11}}{\sum_{i=1}^{k} n_i(n_i - 1)/2}, \quad W_2(C_1, C_2) \;=\; \frac{N_{11}}{\sum_{j=1}^{l} n_j(n_j - 1)/2}$$

where n_i stands for the size of the i-th element of C_1 and n_j the j-th element of C_2. The terms W_1 and W_2 represent the probability that a pair of points which are in the same cluster under C_1 are also in the same cluster under C_2 and vice versa.

Finally, the Jacard index [1] is given by

$$\mathcal{J}(C_1, C_2) \;=\; \frac{N_{11}}{N_{11} + N_{10} + N_{01}}$$

The three indices are all similarity measures and take values out of $[0, 1]$. A straightforward transformation, e.g. $1 - \mathcal{R}(C_1, C_2)$, turns them into a distance measure such that a value of zero implies a perfect matching, i.e. two identical clusterings.

At first glance, the computation of N_{11}, N_{00}, N_{10}, and N_{01} is computationally very expensive. A naive approach would need $O(N^4)$ operations when dealing with images of size $N \times N$. Fortunately, we may make use of the confusion matrix, also called association matrix or contingency table, of C_1 and C_2. It is

a $k \times l$ matrix, whose ij-th element m_{ij} represents the number of points in the intersection of c_i of C_1 and c_j of C_2, i.e. $m_{ij} = |c_i \cap c_j|$. It can be shown that

$$N_{11} = \frac{1}{2}\left(\sum_{i=1}^{k}\sum_{j=1}^{l} m_{ij}^2 - n\right) \qquad N_{00} = \frac{1}{2}\left(n^2 - \sum_{i=1}^{k} n_i^2 - \sum_{j=1}^{l} n_j^2 + \sum_{i=1}^{k}\sum_{j=1}^{l} m_{ij}^2\right)$$

$$N_{10} = \frac{1}{2}\left(\sum_{i=1}^{k} n_i^2 - \sum_{i=1}^{k}\sum_{j=1}^{l} m_{ij}^2\right) \quad N_{01} = \frac{1}{2}\left(\sum_{j=1}^{l} n_j^2 - \sum_{i=1}^{k}\sum_{j=1}^{l} m_{ij}^2\right)$$

These relationships make the indices presented above tractable for large-scale clustering problems like image segmentation.

3.2 Distance of Clusterings by Set Matching

This second class of comparison criteria is based on set cardinality alone. Using the term

$$a(C_1, C_2) \;=\; \sum_{c_i \in C_1} \max_{c_j \in C_2} |c_i \cap c_j|$$

Van Dongen [16] proposes the index

$$\mathcal{D}(C_1, C_2) \;=\; 2n - a(C_1, C_2) - a(C_2, C_1)$$

and proves that $\mathcal{D}(C_1, C_2)$ is a metric.

It can be easily shown that this index is related to the performance measure in [6]. The only difference is that the former is a distance (dissimilarity) while the latter is a similarity measure and therefore they can be mapped to each other by a simple linear transformation.

3.3 Information-Theoretic Distance of Clusterings

Mutual information MI is a well-known concept in information theory. It measures how much information about random variable Y is obtained from observing random variable X. Let X and Y be two random variables with joint probability distribution $p(x, y)$ and marginal probability functions $p(x)$ and $p(y)$. Then the mutual information of X and Y, $MI(X, Y)$, is defined as

$$MI(X, Y) \;=\; \sum_{(x,y)} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

In the context of measuring the distance of two clusterings C_1 and C_2 over a set O of objects, the discrete values of random variable X are the different clusters $c_i \in C_1$ an element of O can be assigned to. Similarly, the discrete values of Y are the different clusters $c_j \in C_2$ an object of O can be assigned to. Hence the equation above becomes

$$MI(C_1, C_2) \;=\; \sum_{c_i \in C_1} \sum_{c_j \in C_2} p(c_i, c_j) \log \frac{p(c_i, c_j)}{p(c_i)p(c_j)}$$

Here all the probability terms can be easily computed from the confusion matrix.

Note that no normalization is provided in MI. In the literature there is a normalized version of mutual information [15]

$$\mathcal{NMI}(C_1, C_2) \;=\; 1 - \frac{1}{\log(k \cdot l)} \sum_{c_i \in C_1} \sum_{c_j \in C_2} p(c_i, c_j) \log \frac{p(c_i, c_j)}{p(c_i)p(c_j)}$$

Meila [11] suggests a further alternative called variation of information

$$\mathcal{VI}(C_1, C_2) \;=\; H(C_1) + H(C_2) - 2MI(C_1, C_2)$$

where

$$H(C_1) \;=\; -\sum_{c_i \in C_1} p(c_i) \log(c_i), \quad H(C_2) \;=\; -\sum_{c_j \in C_2} p(c_j) \log(c_j)$$

represent the entropy of C_1 and C_2, respectively. This index turns out to be a metric.

4 Experimental Validation

In the following we present experiments to validate the measures defined in the last section. Some comparison work has also been done. For this purpose we consider the Hoover measure [5]. The measure from [6] was ignored because of its equivalence to the van Dongen index and the measure from [10] due to its insensitivity to over- and under-segmentation.

For the sake of clarity we consistently transformed all indices into distance measures, implying that a value of zero implies a perfect matching, i.e. two identical clusterings. Among the five performance measures from [5] we only consider the correct detection CD. The transformation $1 - \frac{CD}{\#\text{RT regions}}$ then turns it into a distance measure.

4.1 Validation on Synthetic Data

The range image sets reported in [5,13] have become popular for evaluating range image segmentation algorithms. Totally, three image sets with manually specified ground truth are available: ABW and Perceptron for planar surfaces and K2T for curved surfaces. For each GT image we constructed several synthetic MS results in the following way. A point p is selected randomly. We find the point q nearest to p which does not belong to the same region as p. Then, q is switched to the region of p provided this step will not produce additional regions. This basic operation is repeated for some $d\%$ of all points. Figure 1 shows one of the ABW GT image and two generated MS versions with different distortion levels.

One may expect that the Hoover index (correct detection) monotonically increases, i.e. becomes worse, with increasing tolerance threshold T. However, this is not true, as illustrated in Table 1 which lists the Hoover index for a

Fig. 1. An ABW image: GT (left) and two synthetic MS versions (middle: 5%, right: 50% distortion)

Table 1. Hoover index for an ABW image. The two instances of inconsistency at 20% and 30% distortion level, respectively, are underlined.

	T=0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
20% distortion	0.778	<u>0.667</u>	<u>0.667</u>	<u>0.667</u>	<u>0.667</u>	0.778	0.778	0.778	1.000	1.000
30%	0.778	0.778	0.778	0.889	0.889	0.889	<u>0.778</u>	0.889	1.000	1.000
40%	0.889	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

particular ABW image as a function of T and the distortion level d. There are two instances of inconsistencies. At distortion level 30%, for example, the index value 0.778 for $T = 0.85$ is lower than 0.889 for $T = 0.80$. Since we usually choose a particular value of T in practice, this kind of inconsistency may cause some unexpected effects in comparing different algorithms.

Another inherent problem of the Hoover index is its insensitivity to distortion. Basically, this index counts the number of correctly detected regions. Increasing distortion level has no influence on the count at all as far as the tolerance threshold T does not become effective. For $T = 0.85$, for instance, the Hoover index remains unchanged (0.778) at both distortion level 20% and 30%. Objectively, however, a significant difference is visible and should be reflected in the performance measures. Obviously, the Hoover index does not perform as one would expect here.

By definition the indices introduced in the last section have a high sensitivity to distortion. Table 2 lists the average values for all thirty ABW test images[2]. Obviously, no inconsistencies occur here and the values are strict monotonically increasing with a growing amount of distortion.

Experiments have also been conducted using the Perceptron image set and we observed similar behavior of the indices. So far, the K2T image set was not tested yet, but we do not expect diverging outcome.

[2] The ABW image set contains forty images and is divided into ten training images and thirty test images. Only the test images were used in our experiments.

Table 2. Comparison of synthetic MS at various distortion levels with GT: Average index values for thirty ABW test images.

	d=5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
$\mathcal{R}(C_1, C_2)$	0.024	0.041	0.055	0.068	0.080	0.091	0.102	0.111	0.120	0.129
$\mathcal{D}(C_1, C_2)$	0.027	0.046	0.063	0.078	0.092	0.105	0.117	0.128	0.138	0.149
$\mathcal{VI}(C_1, C_2)$	0.392	0.601	0.758	0.888	1.002	1.099	1.186	1.260	1.329	1.390

Table 3. Index values for thirty ABW test images

	$\mathcal{R}(C_1, C_2)$	$\mathcal{F}(C_1, C_2)$	$\mathcal{J}(C_1, C_2)$	$\mathcal{D}(C_1, C_2)$	$\mathcal{NMI}(C_1, C_2)$	$\mathcal{VI}(C_1, C_2)$	Hoover
UE	0.005	0.010	0.020	0.009	0.707	0.147	0.122
UB	0.008	0.016	0.031	0.013	0.714	0.209	0.180
USF	0.008	0.017	0.033	0.015	0.711	0.224	0.230
UW	0.009	0.017	0.033	0.019	0.848	0.236	0.435

4.2 Validation on Real Data

In [5] four segmentation algorithms have been evaluated using the Hoover measures: UE (University of Edinburgh), UB (University of Bern), USF (University of South Florida) and UW (University of Washington). Table 3 reports an evaluation of these algorithms by means of the indices introduced in this paper. The results imply a ranking of segmentation quality: UE, UB, USF, UW which coincides well with the ranking from the Hoover index (compare the Hoover index values in Table 3 and the original work [5]). Note that the comments above on Perceptron and K2T image set apply here as well.

5 Conclusions

Considering image segmentation as a task of data clustering opens the door for a variety of measures which are not known/popular in computer vision. In this paper we have presented several indices developed in the statistics and learning community. Some of them are even metrics. Experimental results have demonstrated favorable behavior of these indices compared to the Hoover measure.

Note that although experimental validation was only done in range image domain, the proposed approach is applicable in any task of segmentation performance evaluation. This includes different imaging modalities (intensity, range, etc.) and different segmentation tasks (surface patches in range images, texture regions in greylevel or color images). In addition the usefulness of these measures is not limited to evaluating different segmentation algorithms. They can also be applied to train the parameters of a single segmentation algorithm [3,12].

Given some reasonable performance measures, we are faced with the problem of choosing a particular one in an evaluation task. Here it is important to realize that the performance measures may be themselves biased in certain situations.

Instead of using a single measure we may take a collection of measures and define an overall performance measure. This way it is more likely to achieve a better behavior by avoiding the bias of the individual measures. The performance measures presented in this paper provide candidates for this approach.

References

1. A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. Proc. of Pacific Symposium on Biocomputing, 6–17, 2002.
2. K.I. Chang, K.W. Bowyer, and M. Sivagurunath. Evaluation of texture segmentation algorithms. Proc. of CVPR, 294–299, 1999.
3. L. Cingue, R. Cucciara, S. Levialdi, S. Martinez, and G. Pignalberi. Optimal range segmentation parameters through genetic algorithms. Proc. of 15th ICPR, Vol. 1, 474–477, Barcelona, 2000.
4. E.B. Fowlkes and C.L. Mallows. A method for comparing two hierarchical clusterings. Journal of the American Statistical Association, 78:553–569, 1983.
5. A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher. An experimental comparison of range image segmentation algorithms. IEEE Trans. on PAMI, 18(7): 673–689, 1996.
6. Q. Huang and B. Dom. Quantitative methods of evaluating image segmentation. Proc. of ICIP, 53–56, 1995.
7. X. Jiang, K. Bowyer, Y. Morioka, S. Hiura, K. Sato, S. Inokuchi, M. Bock, C. Guerra, R.E. Loke, and J.M.H. du Buf. Some further results of experimental comparison of range image segmentation algorithms. Proc. of 15th ICPR, Vol. 4, 877–881, Barcelona, 2000.
8. X. Jiang.An adaptive contour closure algorithm and its experimental evaluation. IEEE Trans. on PAMI, 22(11): 1252–1265, 2000.
9. X. Jiang. Performance evaluation of image segmentation algorithms. In: Handbook of Pattern Recognition and Computer Vision (C.H. Chen and P.S.P. Wang, Eds.), 3rd Edition. World Scientific, 525–542, 2005.
10. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its applications to evaluating segmentation algorithms and measuring ecological statistics. Proc. of ICCV, Vol. 2, 416–423, 2001.
11. M. Meila. Comparing clusterings by the variation of information. Proc. of 6th Annual Conference on Learning Theory, 2003.
12. J. Min, M. Powell, and K.W. Bowyer. Automated performance evaluation of range image segmentation algorithms. IEEE Trans. on SMC-B, 34(1): 263–271, 2004.
13. M.W. Powell, K.W. Bowyer, X. Jiang, and H. Bunke. Comparing curved-surface range image segmenters. Proc. of 6th ICCV, 286–291, Bombay, 1998.
14. W.M. Rand. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66:846–850, 1971.
15. A. Strehl, J. Gosh, and R. Mooney. Impact of similarity measures on web-page clustering. Proc. of AAAI Workshop of Artificial Intelligence for Web Search, 58–64, 2000.
16. S. van Dongen. Performance criteria for graph clustering and Markov cluster experiments. Technical Report INS-R0012, Centrum voor Wiskunde en Informatica, 2000.

Image Segmentation Based on Genetic Algorithms Combination

Vito Di Gesù and Giosuè Lo Bosco

DMA, Università di Palermo, Italy

Abstract. The paper describes a new image segmentation algorithm called *Combined Genetic segmentation* which is based on a genetic algorithm. Here, the segmentation is considered as a clustering of pixels and a similarity function based on spatial and intensity pixel features is used. The proposed methodology starts from the assumption that an image segmentation problem can be treated as a *Global Optimization Problem*. The results of the image segmentations algorithm has been compared with recent existing techniques. Several experiments, performed on real images, show good performances of our approach compared to other existing methods.

1 Introduction

The segmentation of images in *homogeneous* components plays a key role in each recognition system so that its accuracy may influence the performance of the overall recognition procedure. On the other hand, image segmentation depends on the context and it is subjective; the decision process is driven by the goal or the purpose of the specific task considered. The image segmentation problem can be formulated as a clustering based on region properties, in this case, one of the problems is to select more compact features space to allow a better separation between regions. An overview of clustering-based methods can be found in [1]. Shi and Malik [2] have considered a $2D$ image segmentation as a *Graph Partitioning Problem* (GPP) solved by a normalized cut criterion. The method finds approximated solutions by solving a generalized eigenvalue system. Here, we will consider the problem of extracting largest image regions that satisfy uniformity conditions in the *intensity/spatial* domains. The segmentation method here described incorporates and generalizes the approach to image segmentation by genetic algorithm described in [3]. The general design of the proposed segmentation procedure, named in the following *Combined Genetic Segmentation* (CGS), is sketched in Figure 1. The procedure includes two phases. In the first phase, the *Global Optimization Phase* (GOP) the segmentation algorithm *Unsupervised Tree Segmentation* (UTS) is applied. UTS is based on a genetic algorithm called *Genetic Segmentation Procedure* (GSP) that segments the image into a fixed number of regions. Here, the largest and uniform, but not necessarily connected, regions (*candidate segments*) are found. Candidate segments are represented by a graph SG that is described in section 5. In the second phase *Maximal Connected Components* (MCC) are computed from the SG. This phase is realized by the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 352–359, 2005.

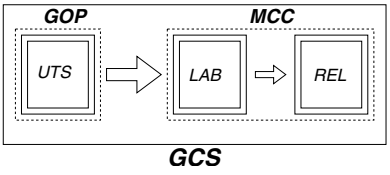


Fig. 1. Sketch of the two phases segmentation algorithm

procedure LAB that updates the labels of the nodes of the SG in order to reach a one to one correspondence between connected components and segments. After the LAB procedure, a generalized relaxation procedure (REL) acts on the SG nodes to embed small components into adjacent largest ones. Section 2 describes the similarity measure between pixels, section 3 gives details on the genetic optimization procedures GSP, section 4 describes the UTS procedure, section 5 describes the algorithm to compute the maximal connected component via the MCC, the generalized relaxation labeling (REL) and the LAB; in section 6 results of the algorithm on real data are shown; concluding remarks are given in section 7.

2 The Similarity Function

In this paper we use a similarity function between two pixels $\underline{x}, \underline{y} \in \mathbf{X}$, that is function of both intensity and spatial features:

$$\delta(\underline{x}, \underline{y}) = a \times d_g(\underline{x}, \underline{y}) + b \times d_E(\underline{x}, \underline{y}) \tag{1}$$

where, $a \geq 0$, $b \geq 0$, and $a + b = 1$. The two terms are defined as follows:

$$d_g(\underline{x}, \underline{y}) = \frac{|g_{\underline{x}} - g_{\underline{y}}|}{max(g_{\underline{x}}, g_{\underline{y}})}, \qquad d_E(\underline{x}, \underline{y}) = \frac{1}{|C_r(\underline{x})|} \sum_{\underline{z} \in C_r(\underline{x}), z \neq x} \frac{|g_{\underline{z}} - g_{\underline{y}}|}{max(g_{\underline{z}}, g_{\underline{y}})} \tag{2}$$

where $C_r(\underline{x})$ is the neighborhood of the pixel $\underline{x}$ and r is the interaction radius. The assumption that pixels may interact at a given distance r is motivated mainly by physiological reasons and it has been also used in [2]; moreover, it allows us to reduce the algorithm complexity.

The component d_g is a metric and it takes into account the closeness of the pixels intensity, while the component d_E considers the attraction effect due to local pixels. Actually, it increases if the intensities of neighbors of $\underline{x}$ differ from the intensity of $\underline{y}$. Note that, d_E is not a metric in fact it is easy to show that $d_E(\underline{x}, \underline{y}) \neq d_E(\underline{y}, \underline{x})$ because in general $C_r(\underline{x}) \neq C_r(\underline{y})$. Therefore, δ is a distance functions only if $b = 0$.

Fig. 2. Chromosome representation of a pixel

3 Genetic Segmentation Procedure

The formalization of the image segmentation problem by different perspective (e.g. as clustering, as graph partitioning) consists in solving it as a global optimization problem which is NP complete [2,4,5].

Genetic algorithms (*GA*'s) [6,7] are random algorithms that can often find a global optimal solution; however, local optima could be reached in consideration of their random nature. They have been already used to face clustering problems [8].

The GSP groups the input data in K segments, it consists in the assignment procedure of each pixel $\underline{x}$ to a given segment class chosen from a random population of assignments. GSP is the kernel of the UTS segmentation procedures described in section 4.

One of the most important step of a genetic algorithm is the data coding that may influence its efficiency (precision of the solutions, computational complexity). In our case, each pixel, (i_x, j_x, g_x), of $\mathbf{X}$, is coded by a binary string (the *chromosome*) α_{i_x,j_x} of 32 bits that represent position (i_x, j_x) in the 24 most significant bits and the pixel-label, λ_x, in the 8 least significant bits. The value λ_x identifies the cluster where x belongs. The genetic operators are applied to the entire binary content of a chromosome and this corresponds to some modification on the binary content part which represents a pixel position or a label. The resulting chromosome could identify a new label assignment for the same pixel (in the case only the least significant bits is modified) or a label assignment for a different pixel. This corresponds to a whole modification of the grouping. The functions used to code the pixel label and positions are:

$$L(\lambda_x) = (\frac{2^8 - 1}{K}) \times \lambda_x \qquad S(k_x) = (\frac{2^{24} - 1}{n \times m}) \times k_x \tag{3}$$

where $k_x = i_x \times m + j_x$ and K is the number of clusters. It follows that $\alpha_{i_x,j_x} \equiv \alpha_{k_x} \equiv b_{31}b_{30}...b_0$, with $b_{31}b_{30}....b_8 = S(k_x)$ and $b_7b_6....b_0 = L(\lambda_x)$ (See Figure 2). A generic chromosome α, represents a pixel in position $S^{-1}(\alpha)$, assigned to a cluster $\lambda = L^{-1}(\alpha)$.

The chosen coding is compact and easy to handle, because it represents the information in four bytes; moreover, it allows us to represent up to 256 segments per image and image-size up to 4096×4096. These values are adequate in most real applications. The accuracy of the approximated solution of the global optimization problem depends also on the choice of the *fitness function*. In our case, the optimization is related to the minimization of the internal variance of each image segment. The evolution of the GA will be driven by a *fitness function*, f, that is computed on the basis of the similarity function, δ, defined

in the previous section. Each segment P_j is characterized by the mean value, mv_j, of the gray levels:

$$mv_j = \frac{\sum_{\alpha \in A_j} \mathbf{X}(S^{-1}(\alpha)))}{|P_j|} \tag{4}$$

where $A_j = \{\alpha | L^{-1}(\alpha) = j\}$. The fitness function is computed between a given chromosome $\alpha = (S(k), L(\lambda))$ and the corresponding segment P_λ as follows:

$$f(\alpha) = \delta(\mathbf{X}(S^{-1}(\alpha)), mv_{L^{-1}(\alpha)}) \tag{5}$$

To evolve the system the single point crossover and the bit mutation are used and their combination is denoted by Γ. Random labels in $\{0, 1, ..., K\}$ are assigned to the initial population of chromosomes. The population at the iteration t is denoted by $P(t) = \{\alpha_1(t), \alpha_2(t), ..., \alpha_N(t)\}$ where $N = n \times m$ is the size of the input image. The application of Γ generates the population $\Gamma(P(t)) = \{\beta_1(t), \beta_2(t), ..., \beta_N(t)\}$ where $\beta_r = \Gamma(\alpha_r)$. The *selection process* is performed by selecting for each $1 \leq r \leq N$ the best chromosome between α_r and β_r. Γ and the selection process are applied until the *halting condition* $|Var_{t-1} - Var_t| \leq \phi$ is satisfied, where $Var_t = \sum_k^K \sigma_t(k)$ is the total internal variance, $\sigma_t(k)$ is the variance of the cluster k at the iteration t and $\phi \geq 0$. The condition $|Var_{t-1} - Var_t| = 0$ is not usually reached and the value of ϕ is determined by the heuristics $\phi \approx \varepsilon \times Var_{t-1}$, with $\varepsilon \leq 0.1$. Finally, the halting condition takes also into account if a maximum of iterations T has been reached. From previous definitions it follows that the genetic algorithm GSP may be sketched as it follows:

Procedure $GSP((S_1, S_2, \ldots, S_K), T, p_c, p_m)$
Set up a population of chromosome $P(0) = \{\alpha_1(0), \alpha_2(0), ..., \alpha_N(0)\}$
by using $(S_1, S_2, \ldots, S_K)$;
t=0;
repeat
Apply Γ to current population $P(t)$;
Build population $P(t+1)$ by selecting the *best chromosome*
from $P(t)$ and $\Gamma(P(t))$;
t = t + 1;
until $(|Var_{t-1} - Var_t| \leq \phi) \bigvee (t > T)$;
Set up $(S_1, S_2, \ldots, S_K)$ by using $P(t-1)$;

Note that the sets $(S_1, S_2, \ldots, S_K)$ represent an initial distribution of labels (segments) that could be user defined or totally randomized. The fitness function used in the GSP is that defined by Equation 5. Parameters a and b determine the influence of the pixel intensity and the local distribution of labels. Probabilities p_c and p_m establish the frequency of the *crossover* and *mutation* operators. Typical values used in the our experiments are $p_c = 0.7$ and $p_m = 0.01$. K indicates the initial maximum number of clusters and it may be reduced from the genetic process.

4 Unsupervised Tree Segmentation

The *UTS* algorithm builds a balanced binary tree, where the apex represents the whole image, **X**, internal nodes are temporary segments, and leaves are final image segments. The UTS uses the genetic procedure *GSP* in the sense that on each internal node of the tree it is applied with $K = 2$. The binary split of each node continues until an internal *uniform condition* is reached or the tree has reached the maximum depth $log_2(K_{max})$ where K_{max} is the maximum number of clusters. The *UTS* algorithm can be sketched as it follows:

Procedure UTS($\mathbf{S}$,K_{max},d)
  if $\neg uniform(\mathbf{S}) \bigwedge (d < log_2(K_{max}))$ **then**
    split randomly $\mathbf{S}$ in (S', S'');
    $GSP((S', S''), T, pc, pm)$;
    $UTS(S', d+1)$;
    $UTS(S'', d+1)$;
  else
    $K = K + 1$;
    $S_K = S$;
  end

The procedure *uniform*($\mathbf{S}$) returns a value that is true if the variance of the gray levels in the set $\mathbf{S}$ is greater than a given threshold. In most cases the threshold is set to $\overline{g} + \sqrt{\overline{g}}$; where $\overline{g}$ is the mean gray value of the pixels in A. This assumption is a good approximation in the case of Poisson's distribution. The computed segments $(S_1, S_2, \ldots, S_K)$ are represented by the leaf nodes on the tree.

5 Maximal Connected Components Computation

The *GOP* procedure assigns a label to each pixel of **X**. The resulting segmentation can be represented as a connected planar labelled graph, named the segmentation graph (*SG*). Nodes of *SG* correspond to connected homogeneous regions (pixels in the same segment); the pair (λ, w), assigned to each node, represents the segment-label (λ) and the number of pixels (w) in the corresponding

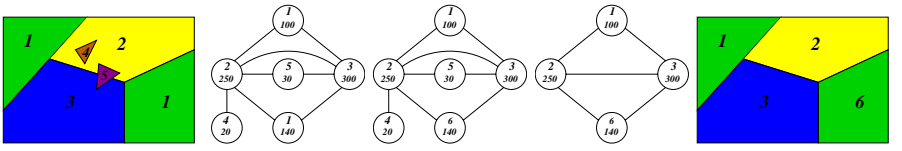


Fig. 3. (a) *GOP* segmentation; (b) corresponding *SG*; (c) *SG* after the application of *LAB* procedure; (d) *SG* after the application of *REL* procedure; (e) final segmentation.

connected region. Note that, segments constituted by disconnected regions corresponds to more than one node with the same λ, but in general different w. However, two adjacent nodes must have different λ, because they represent, by construction, adjacent regions that can't correspond to the same segment. The Maximal Connected Component (MCC) algorithm is performed in two phases:

- LAB. During this phase, all λ's are recomputed to assign a different label to each node. After this phase each node of the SG corresponds to a maximal connected region of homogeneous pixels.
- REL. During this phase, nodes of SG, such that $w < \psi$, are embedded into adjacent greater nodes, in particular the small nodes are embedded in the neighbor which has the maximum cardinality greater than ψ all over the neighborhood of w. The threshold can be set to $\psi = \overline{w} - \sqrt{\overline{w}}$; where $\overline{w}$ is the mean value of the node weights. As a result small segments are eliminated.

For example, Figure 3a represents the result of a segmentation after UTS, Figure 3b represents the corresponding graph, SG. Figure 3c shows SG after the LAB procedure, finally Figure 3d shows SG after the application of REL. Figure 3e shows the final segmentation.

6 Experimental Results

The experiments has been performed on images belonging to Corel data set (http://elib.cs.berkeley.edu/photos/corel/) in order to compare the accuracy of CGS with other three segmentation algorithms : the first based on a C-means clustering (*C-means*), the second on a single link clustering (*Single-link*) and the last on a normalized cut minimization criterion (GPS). The first two algorithm has been implemented on the R.S.I.S. [9] image analysis system, the last one is the method by Shi and Malik [2]. In the following we provide segmentation examples of four images: church (192×128 pixels), face (128×192), bear (128×192) and fragments (128×192). The results of CGS, *C-means*, *Single-link*, and GPS segmentation techniques on the test images are reported in Figure 4.

The performance evaluation of segmentation methods is not an easy task because the expected results are subjective and they depend on the application. One evaluation procedure could be the comparison of the method against a robust and well experimented one, but this choice is not always feasible. Here, we evaluate the performances of all the algorithm by comparing machine and human segmentation. The comparison has been performed between the automatic segmentation and the segmentation deriving from the evaluation of an odd number (5) of persons. The human segmentation is obtained taking the maximum intersection of the regions selected by each person separately. This procedure has been suggested by the approach described and tested experimentally in [10]. In the following, Seg_k and S denote the *k-th* segment retrieved by humans and machine respectively, $|Seg_k|$ and $|S|$ denote the corresponding size, $\sharp \text{agr}_k$ is the largest pixel intersection between Seg_k and S. A measure of agreement between human and automatic segmentation can been evaluated as follows:

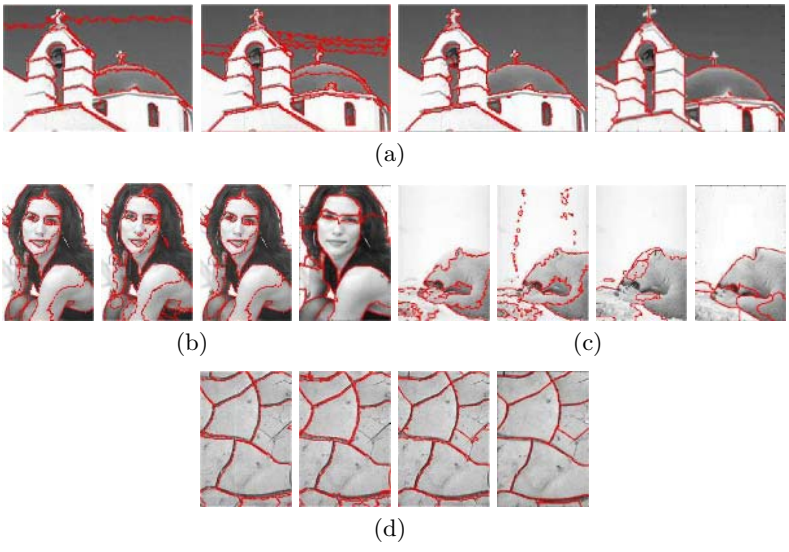


Fig. 4. For each image the different segmentation results, from left to right : *CGS*; *C-means*; *Single-link*; *GPS*

Table 1. Segmentation Comparison

Method	η	T_{CPU}
		sec.
CGS	0.74	154
C-means	0.70	97
Single-link	0, 77	220
GPS	0, 73	300

$$\eta = \frac{1}{K}\sum_{k=1}^{K}\beta_k \times \frac{\sharp \text{agr}_k}{max(|Seg_k|, |S|)}$$

where β_k is the percentage of pixels of the whole image included in the k-th segment. This measure error is related to the global consistence error introduced in [10]. Table 1 shows the average value of the agreement indicator (η) and the mean CPU time (T_{CPU}) for each segmentation method. All the segmentation algorithms have been implemented in C++ under windows system; the CPU time is referred to an INTEL PENTIUM III 1GHz. The results in the table show that all algorithms provide a comparable accuracy. *Single-link* has the best accuracy, while *CGS* provides a good compromise in time/accuracy. It must be pointed out that *GPS* [2] is more complete since it is also able to segment texture based images. The convergence of the *CGS* depends on the convergence of the components *GSP*. Their convergence has not been fully proved; nevertheless the *GA* converges in all our experiments.

7 Concluding Remarks

The image segmentation problem has been treated here as a complex global optimization problem that, because of its computational complexity, does not admit tractable exact solutions. Thus, we have proposed a segmentation algorithm based on a genetic algorithm to find an approximated solution of the problem. The comparison with three published algorithms shows its good performance. The agreement between natural and automatic grouping has been also done using more than one individual. Finally, the overall method seems to indicate a faster convergence to the correct solution.

References

1. A.K. Jain and P. Flynn, "Image Segmentation Using Clustering", *Advances In Image Understanding*: A Festschrift for Azriel Rosenfeld, K. Bowyer and N. Ahuja (eds.), pp. 65–83, IEEE Computer Society Press, 1996.
2. J. Shi, J. Malik, "Normalized Cuts and Image Segmentation", *IEEE transactions on PAMI*, Vol.22, N.8, pp.1–18, IEEE Computer Society Press, 2000.
3. G. Lo Bosco, "A genetic algorithm for image segmentation", *in Proc. of ICIAP 2001*, pp. 262–266, Palermo, Italy, IEEE computer society press, 2001.
4. M. Garey, D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", W.H. Freeman and Company, 1979.
5. R. Horst and P.M. Pardalos (eds.), Handbook of Global Optimization, Kluwer Academic Publishers, 1995.
6. J.H. Holland, "Adaptation in natural and artificial systems", University of Michigan Press, Ann Arbor, MI, 1975.
7. D.E.Goldberg, "Genetic algorithms in search, optimization and machine leraring", Addison Wesley, 1989.
8. L.O. Hall, I.B. Ozyurt, J.C. Bezdek, "Clustering with a genetically optimized approach", *IEEE Transactions on Evolutionary Computation*, Vol.3, N.2, pp. 103–112, IEEE Computer Society Press, 1999.
9. R.S.I.S., EPRI TR-11838, WO-5144-03 & WO-8632-01, Palo Alto, October, 1999.
10. D. Martin, C. Fowlkes, D. Tal, J. Malik. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms", *in Proc. of ICCV01*, Vol.2, pp. 416–425, IEEE Computer Society press, 2001.

Image Segmentation Through Dual Pyramid of Agents

K. Idir, H. Merouani, and Y. Tlili

Laboratory of computer science Research., Pattern Recognition Group,
Dept. of computer science – Faculty of engineer science,
Badji Mokhtar University, BP.12-23200, Annaba, Algeria
Karima_id2@yahoo.fr, hayet_merouani@yahoo.fr, guiyam@yahoo.fr

Abstract An effective method for the early detection of breast cancer is the mammographic screening. One of the most important signs of early breast cancer is the presence of microcalcifications. For the detection of microcalcification in a mammography image, we propose to conceive a multi-agent system based on a dual irregular pyramid.

An initial segmentation is obtained by an incremental approach; the result represents level zero of the pyramid. The edge information obtained by application of the Canny filter is taken into account to affine the segmentation. The edge-agents and region-agents cooper level by level of the pyramid by exploiting its various characteristics to provide the segmentation process convergence.

Keywords: Dual Pyramid, Image Segmentation, Multi-agent System, Region/Edge Cooperation.

1 Introduction

Image segmentation is one of the most widely used steps in the process of reducing images to get useful information. It allows the partitioning of image into several distinct regions.

We propose for the detection of microcalcification in a mammography image, to cooperate region-agents obtained by an incremental approach and edge-agents obtained by a Canny filter in a pyramidal structure.

Initially, a short outline of the irregular pyramids is presented, then we expose in brief way some work carried out in image segmentation by multi-agent approach, there after we present our approach while arguing the choice of the dual pyramid and the role of the agents. Finally we conclude this paper by some prospects for possible improvements.

2 Irregular Pyramid

A pyramid is a stack of images with decreasing resolutions; the bottom level of a pyramid is the original image. Each level of the pyramid can be described by the neighborhood graph where the set of vertices correspond to regions of the level and

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 360 – 366, 2005.

the edges' set correspond to the adjacency relations between regions. The algorithm for the pyramid construction proceeds in the following steps [1]:

- Creation of level zero (base), starting from image to be treated.
- Construction of the similarity graph starting from the adjacency graph. In order to hold account of the contents of image, each vertex is connected only to vertices, which are declared to him similar.
- The graph of the k level is obtained by processing decimation on the graph of k-1 level. This decimation procedure is based on two rules.
  1. Two neighbors at a given level cannot both survive at the next level;
  2. For each non-surviving cell, there is at least one surviving cell in its neighborhood.

  The decimation can be:
  - Stochastic: [11] the choice of the surviving vertices is done randomly.
  - Adaptative: [9] a vertex survives if it locally minimizes an *interest operator* based on the variance of the region associated to the vertex.
- Assignment of each non survivor vertex to one survivor parent.
- Attributes of each surviving vertex are calculated according to the attributes of vertices of the level below that represent.
- Construction of the neighborhood graph of the new level.

The irregular pyramids operate on general graph structures instead of the regular neighborhoods as in the case of regular pyramids (example: "Quadtree"). The cells of one level represent each one a region of which the form is not constrained by a geometrical pattern (square, triangle, polygon). The result obtained by the application of the irregular pyramids is independent of the course of image.

However these pyramidal structures can generate a non-existent borders or forget existing discontinuities, to avoid this disadvantage, cooperative approaches are proposed. For a natural aspect of cooperation between region and edge multi-agent systems are well adapted to develop such approaches.

3 Multi-agent Approaches in Image Segmentation

According to the application developed in previous works, different kind of agents with a variety of characteristics, interaction and coordination concepts can be found.

One of the aspects of Boucher system [2] is the manner of launching agents in the next image of the sequence according to what is being held in current image. The agents thus launched start by segmenting their components and will meet progressively while growing. Two agents which meet and discover that work both on the same component of image can fuse their efforts in order to decrease the number of redundant agent. The agents used for the study of cytological images sequences, have four behaviors: *The perception behavior* to evaluate pixels, *the interaction behavior* to manage fusions, *the differentiation behavior* to evaluate the primitive in order to determinate the kind of component is closed to and *the reproduction behavior* to create agents at certain places of image.

In a context of MRI segmentation, Germond [7] cooper edge-agents based on (A*) algorithm and region-growing agents specialized for the detection of the white matter

(WM) and the grey matter (GM) of the brain. The system comprises a deformable model that provides valuable information on the brain boundary, to position the GM-agents that allow eventually the localization of WM-agents. Edge-agents are used to refine brain boundary assessed by the deformable model.

Duchesnay [5] presents a society of agents (region-agent, edge-agent) organized in irregular pyramid while detailing the initialization process of the pyramid and the total control which intervenes in the passage of one level to another. An agent of the pyramid stands for a region primitive (obtained by the Quadtree) method or edge primitive, he will supplement this primitive while merging with others agents by running a sequence of seven behaviors:

1. *Territory marking and features extraction:* each agent marks a territory that corresponds to its primitive
2. *Exploration:* Agents are exploring a shared environment around their territory to discover their neighbors.
3. *Merging planning:* each region agent tries to find out (similar) neighbors to merge with.
4. *Cooperation:* agents cooperate with each other to enhance the quality of their merging plan.
5. *Decimation:* the number of agents has to decrease level by level, then a distributed decimation protocol, performed by the agents, selects survivors.
6. *Reproduction:* each survivor agent creates a new agent in the next level of the pyramid.
7. *Attachment:* all agents should be represented in the next level, then each non-surviving agent looks for the best representative in its neighborhood.

Region-agents and edge-agents cooperate for the segmentation of MRI, in a multi-agent plate forms proposed by Settache [12]. Two images are inputted to the multi-agent system, region-chart (obtained by Quadtree algorithm) and edge-chart (obtained by a Deriche filter, then chaining of edges), from which region and edge agents are launched. Segmentation-agents, whose behaviors are defined by an automat related on the region-agents and the edge-agents, are responsible for the improvement of the initial pre-segmentation.

4 Proposed Approach

We propose to conceive a multi-agent system for image segmentation, which allows a cooperation region/edge, and it differs primarily from the work presented in Sect. 3 by the use of dual pyramid.

4.1 Region Map

The growing region method (pixel aggregation process) is chosen for pre-segmenting the image into regions. This approach preserves the form of each region. To build a "region" primitive, first select a special set of pixels called *seeds*. Regions are grown by aggregating to the seeds pixels which verify a fixed criterion. Thus, a membership of a pixel into region taking into account: information of region and local information relating to the pixel.

The evaluation of a pixel to annex to a region depends on two weighted homogeneity criteria: average of gray level and the variance of region. The image result is the base of our pyramid, where each region will be represented by an agent.

4.2 Edge Map

To affine segmentation, we propose to hold account the edge information, obtained by applying the Canny filter [3].

4.3 The Dual Pyramid

The use of simple graphs to the irregular pyramids forces to have only one edge between two vertices in the graph, thus, it is unable to distinguish from the graph an adjacency relation from inclusion relation between two regions. This limitation was raised by Kropatsch [10], Willersinn [13] and others who propose to build a dual irregular pyramid complementing each level of an irregular pyramid by its dual graph during construction. The dual graph preserves the topology of the adjacencies and correctly codes the inclusion relation between regions.

The basic process for construction of the dual pyramid is the dual decimation.

Decimation Parameters. G_i (V_i, E_i) a planar graph, $S_i \subset V_i$ a subset of surviving vertices and $N_{i,\, i+1} \subset E_i$ a subset of primary non-surviving edges. The couple (S_i, $N_{i,i+1}$) called *decimation parameters* determine the structure of an irregular pyramid. Every non-surviving vertex *(Child)* must be connected to one surviving vertex *(Parent*) in a unique way. The selection of the surviving vertices can be done according to rules of decimation mentioned in Sect. 2.

Dual Decimation. Dual decimation combines the selection of decimation parameters and the dual contraction that proceeds in two basic steps dual edge contraction and dual face contraction. (Fig. 1)

1. *Selection of the decimation parameters (S_i, $N_{i,i+1}$):*
Identification of the survivors and non-survivors.

2. *Dual edge contraction* $N_{i,i+1}$*:*
The contraction of a primary non-surviving edge from $N_{i,i+1}$, consists in the identification of its endpoints (vertices) and the removal of both the contracted edge and its dual edge.

After dual contraction, faces with degree one or two may result. They correspond to self-loop and double edges in the neighborhood graph.

A second (dual) contraction process 'cleans' the dual graph from such degenerated faces but not those enclosing any surviving parts of the graph. They are necessary to preserve correct structure.

3. Dual face contraction:
- Selection of the decimation parameters (S^*_i, $N^*_{i,i+1}$):
- S^*_i, the set of faces with degree above 2.(surviving faces)
- $N^*_{i,i+1}$ denotes the set of edges having one surviving faces as one end vertex and non-surviving faces as the other end vertex.

- Dually contract edge $N^*_{i,i+1}$
- Repeat until all degenerated faces have been eliminated.

The relation between two pairs of dual graphs, $(G_i, \overline{G_i})$ and $(G_{i+1}, \overline{G_{i+1}})$ established by dual graph contraction with decimation parameters $(S_i, N_{i,i+1})$ is expressed by function C

$$(G_{i+1}, \overline{G_{i+1}}) = C\,[(G_i, \overline{G_i}), (S_i, N_{i,i+1})] \quad (1)$$

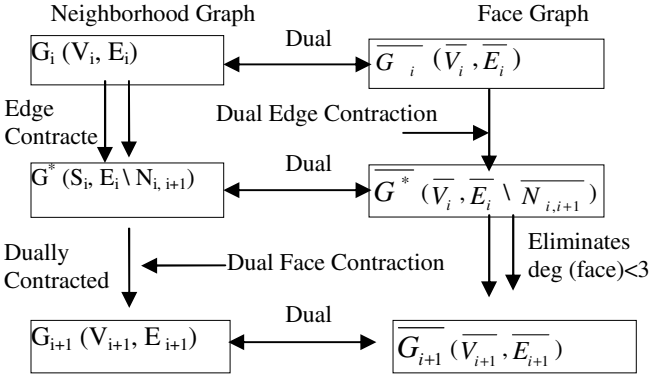
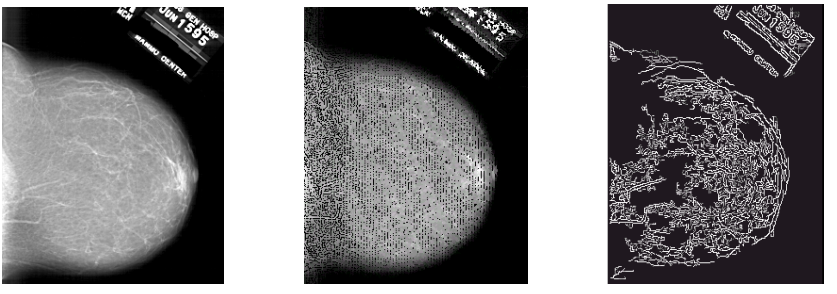


Fig. 1. Dual Graph Contraction: (Extract from ([8]) $(G_{i+1}, \overline{G_{i+1}}) = C\,[(G_i, \overline{G_i}), (S_i, N_{i,i+1})]$

4.4 Dual Pyramid of Agent for the Detection of Microcalcification

An effective method for the early detection of breast cancer is the mammographic screening. One of the most important signs of early breast cancer is the presence of microcalcifications. A microcalcification is a tiny white speck seen on a mammogram. It represents flecks of calcium salts, and is often the only indicators of malignant tumors [6].

For the detection of microcalcification in a mammography image, we propose to conceive a multi-agent system based on pyramidal structure.



(a) Initial Image. *(b) Region map* *(c) Edge map*

Fig. 2. Preprocessing of the image (Extract from [4])

The use of the agents instead of simple vertices enriches the decisional framework of the approach. Indeed the agents provide a concept particularly adapted for the expression of the cooperation and the negotiation between closed regions.

Different kinds of agent are proposed for our system:

- Agents allowing the control and the correct working of the multi-agents system: user interfaces-agents, monitor-agent, sequencer-agent and agent known as dual-agent to control the various steps of the dual decimation.
- A region-agents and edge-agents: Each agent is related to a primitive region from region-map (figure 2b) or edge from edge-map (figure 2c). It has all relevant information about it (examples: average and compactness of the region, length and continuity of edge).
- The interaction for the construction of the pyramid allows two agents provided with certain behaviors, to join together their primitives (region or edge):
- A behavior allowing agent **to mark a territory** corresponding to its primitive (region, edge).
- A behavior allowing agent **to discover its neighbors**. The connection of region agent is restricted to 4 neighbors, to have a planar graph at the base level of the pyramid. For the neighborhoods of the edge-agent we use the distance between the ends of edge.
- A behavior allowing agent **to interact**, **to decide** who **survive** and with which agent he wish **to merge** by calculating a *fusion desire* of his neighborhoods expressed in function of the average of grey level and the variance of regions. A candidate regions list of a region-agent represents regions with *fusion desire* greater than a fixed threshold.
- A behavior allowing agent to confirm a fusion, for example region-agent asks an edge-agent about an edge between two candidates regions to fuse.
- A behavior to select the survivor's agents by application of the rules of an adaptative decimation where the *interest operator* of an agent is the sum of the desire of fusion for all his candidates regions.
- A behavior allowing each **non-survivor** agent to be **represented** by the best **survivor** agent in its neighbors at the next level of the pyramid
- A behavior allowing a surviving agent to **create a new agent** at the **next level** of the pyramid.

To control the process of the dual decimation a dual-agent executes two behaviors:

- A behavior allowing **to build a dual graph**, to extract faces, discover neighborhood of faces, find correspondence between dual graph and adjacency graph.
- A behavior to **control the process**, relating to the **dual contraction** of edges and faces.

5 Conclusion

The image segmentation constitutes the base of the process image interpretation; it remains an active subject of research in artificial vision. Many techniques are

available to deal with the image segmentation problems. In this article, we propose to cooperate region based-method and edge based-method by multi-agent system in a dual pyramid for segmenting mammography image.

This work is in progress, we envisage to implement various modes of cooperation: cooperation region/edge, region/region and edge/edge in order to ensure the convergence of the segmentation process.

References

1. P. Bertolino. Contributions des pyramides irrégulières en segmentation d'images multirésolution. Ph.D. thesis, Institut National Polytechnique de Grenoble, 1995
2. Boucher. Une approche décentralisée et adaptative de la gestion d'informations en vision, Application à l'interprétation d'images de cellules en mouvement. Ph.D. thesis, Université Joseph Fourier, Grenoble, 1999.
3. J.F. Canny. A computational approach to edge detection. *Patt Ana.Mach. Int*, 8(6):679-698, 1986.
4. DDSM: Digital Database for Screening Mammography. University of South Florida. http://marathon.csee.usf.edu/Mammography/Database.html00
5. E. Duchesnay. Agents situés dans l'image et organisés en pyramide irrégulière. Contribution à la segmentation par une approche d'agrégation coopérative et adaptative. Ph.D. Université de Rennes-1, 2001.
6. F. Gaspoz. Mammographie digitale & Analyse d'images par ordinateur. Laboratoire TIMC – IMAG. Faculté de Médecine Grenoble. Université Joseph Fourier Grenoble, France. 2003
7. L. Germond, M.Dojat, C. Taylor, C. Garbay. A Cooperative Framework for Segmentation of MRI Brain Scans. *Artif. Intell. in Med*. 20 (2000) 277-94.
8. Y. Haxhimusa, W.G. Kropatsch. Hierarchical Image Partitioning with Dual Graph Contraction. Technical Report PRIP-TR-81, Institute of Computer Aided Automation 183/2, Patt. Recogn. Image. Proc Group, Austria, 2003
9. J.M. Jolion and A. Montanvert. The adapted pyramid: a framework for 2d image analysis. *Computer Vision Graphics and Image Processing*, 55(3):339-348, May 1992
10. W.G. Kropatsch. Building irregular pyramids by Dual Graph Contraction. Technical Report PRIP-TR-35, Institute of Automation 183/2, Dept. for Patt. Rec. Image. Proc, TU Wien,Austria, 1994
11. P. Meer. Stochastic image pyramids. *Comp. Vision. Graph. Image Proc.*. 45 (3): 269-294. 1989.
12. H. Settache. Une plate-forme multi-agents pour la segmentation d'images: Application dans le domaine des IRM cérébrales 2D. DEA Report, Université de Caen, 2002
13. D. Willersinn. Parallel Graph Contraction for Dual Irregular Pyramids. PRIP-TR 28, Institute for Automation, 183/2, Technical University of Vienna, Austria, 1994

A New Wavelet-Domain HMTseg Algorithm for Remotely Sensed Image Segmentation

Qiang Sun, Biao Hou, and Li-cheng Jiao

Institute of Intelligent Information Processing, Xidian University,
710071 Xi'an, China
art_navigator@yahoo.com.cn

Abstract. A new wavelet-domain HMTseg method is proposed, which fuses the segmentation results at coarse and fine scales with a new and feasible context model together with one preprocessing of raw segmentations at different scales. Compared to the original HMTseg method, the new method not only lays emphasis on the performance from coarse-scale segmentation, preserves the main outlines of the homogeneous regions in an image, and thus achieves good region consistency of segmentation, but also take into account the information from fine-scale segmentation, thus improves the accuracy of boundary localization of segmentation and enables the discrimination of small targets in an image, which is desirable for interpretation of remotely sensed images. Experiments on remotely sensed images, including aerial photos and SAR images, demonstrate that the proposed method can effectively take into consideration both the region consistency and the accuracy of boundary localization of segmentation performance, and give better segmentation results.

1 Introduction

In most recent years, wavelet-domain statistical image models, especially hidden Markov tree (HMT) [1] models, have gained more and more attention from image processing and analysis community due to their effectiveness and flexibility in performing image analysis tasks. Choi et al proposed a new framework, HMTseg [2], for multiscale Bayesian image segmentation based on wavelet-domain HMT models pioneered by Crouse et al to give the statistical characterization of signals by capturing inter-scale dependencies of wavelet coefficients. In HMTseg method, the raw maximum likelihood (ML) segmentations at different scales are yielded before the inter-scale fusion of class labels from coarse scale to fine one. Another tree structure, the context labeling tree (CLT), was designed to exploit the dependencies of parent and child labels for the dyadic image squares across scales. The final classification of each dyadic square at different scales, except for the coarsest one of wavelet decomposition, was implemented in manner of scale recursion. In [2], one simplified context model is used to implement the inter-scale fusion in HMTseg method. This model is typically effective for images mostly made up of homogeneous regions, but fails to perform well for images consisting of complex structured information and/or

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 367 – 374, 2005.

more inhomogeneous regions, remotely sensed images involved in this paper for example. In addition, this segmentation method, suitable for natural or textured images, could not give good results for synthetic aperture radar (SAR) images due to a particular kind of noise, speckle, inherent in them.

In this paper, we propose a modified HMTseg method to consider the region consistency (robust classification) of ML raw segmentation for dyadic squares at coarse scale and the accuracy of boundary localization (poor classification) at fine scale, based on a new and feasible context model taking into account the information from coarser scales and fine ones simultaneously. Meanwhile, a preprocessing stage is also introduced to further amend the raw ML segmentations at different scales to favor better multiscale fusion eventually.

This paper is organized as follows. The HMT method [2] is briefly reviewed in section 2. In section 3, the image segmentation using the modified HMTseg method is detailed. Simulation results and analysis are given in section 4. Finally, one conclusion is drawn and future work directed in section 5.

2 HMTseg Method for Multiscale Image Segmentation

The HMTseg method relies on three separate tree structures: the wavelet transform quad-tree, the HMT, and a labeling tree [2]. As a complete procedure, it includes three essential ingredients, i.e. HMT model training, multiscale likelihood computation, and fusion of multiscale maximum likelihood (ML) raw segmentations.

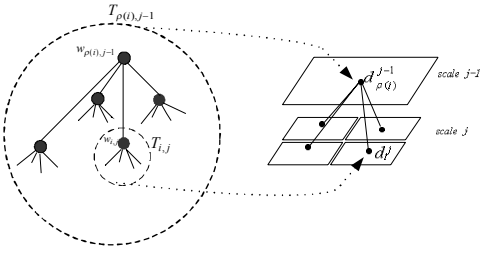


Fig. 1. Correspondence of quad-tree structure of wavelet coefficients with multiscale representation of an image

HMT model training is used to fitting a certain set of model parameters to a given set of training data extracted from a certain homogeneous region in an image. To this end, an iterative procedure, 2-D extension of EM algorithm for 1-D HMT model [1], is exploited to find one locally optimal set $\mathbf{\Theta}_c$ of model parameters for a given set of training data. The likelihood that training data come from a certain model is maximized via a few iterations of E and M steps of the EM algorithm, as a consequence the set $\mathbf{\Theta}_c$ of model parameters that locally maximizes the likelihood can be obtained. To train a pixel brightness probability density function (PDF), Gaussian mixture model (GMM) is used to train each texture and thus to obtain pixel-level segmentations.

As for multiscale likelihood computation, a single upward sweep [a fast $O(n)$ algorithm] through the HMT can deal with it [2]. Each subtree of wavelet coefficients residing in one of the subbands corresponds to one specific dyadic image square at each different scale. For example, the subtree $T_{i;j}$ rooted at $w_{i,j}$ can be brought into correspondence with the dyadic square $\mathbf{d}_i^j$, as illustrated in Fig. 1. Thus, the calculation of likelihood that the dyadic square $\mathbf{d}_i^j$ is produced by the parametric set $\mathbf{\Theta}_c$ can be then transformed into the computation of the likelihood that the coefficient subtree $T_{i;j}$ is generated by $\mathbf{\Theta}_c$. In this way, the likelihoods of all dyadic squares of an image can be obtained in a single upward sweep through the tree of wavelet coefficients since the wavelet transform and HMT possess the similar multiscale structure, as detailed in [2]. In fact, the wavelet coefficients corresponding to the dyadic square $\mathbf{d}_i$ are composed of one triple $\left\{\mathcal{T}_i^{\mathrm{LH}}, \mathcal{T}_i^{\mathrm{HL}}, \mathcal{T}_i^{\mathrm{HH}}\right\}$, each a subtree of one of the three wavelet subband quadtrees. Under the independent subband assumption, the likelihood that the dyadic square $\mathbf{d}_i$ is gained by the model $\mathbf{\Theta}_c$ can be expressed as

$$f(\mathbf{d}_i|\mathbf{\Theta}_c) = f(\mathcal{T}_i^{\mathrm{LH}}|\mathbf{\Theta}_c^{\mathrm{LH}}) f(\mathcal{T}_i^{\mathrm{HL}}|\mathbf{\Theta}_c^{\mathrm{HL}}) f(\mathcal{T}_i^{\mathrm{HH}}|\mathbf{\Theta}_c^{\mathrm{HH}}) . \tag{1}$$

Finally, the ML classification can be obtained by

$$\hat{c}_i^{\mathrm{ML}} := \arg\max_{c\in\{1,2,\ldots,N_c\}} f(\mathbf{d}_i|\mathbf{\Theta}_c) , \tag{2}$$

where N_c is the number of texture classes in an image. For different scale representation of an image in a pyramidal structure, a set of ML segmentations can be obtained in the same way. Hence, this step yields the ML raw segmentations at different scales, which is the basis of the following fusion procedure.

The third step in HMTseg method is to intelligently combine the raw multiscale ML segmentations given at the second step using a Bayesian inter-scale fusion technique. This idea is based on the fact that finer-scale dyadic squares nest inside coarser-scale squares, and the dyadic squares are statistically dependent across scale for images consisting of fairly large, homogeneous regions [2]. During the fusion procedure, relatively reliable coarser-scale information is used to guide less reliable fine-scale decisions. This Bayesian inter-scale decision fusion computes one maximum a posterior (MAP) estimate $\hat{c}_i^{\mathrm{MAP}}$ for the class label of each $\mathbf{d}_i$, i.e.

$$\hat{c}_i^{\mathrm{MAP}} := \arg\max_{c_i\in\{1,2,\ldots,N_c\}} \mathrm{p}(c_i|\mathbf{x}) = \arg\max_{c_i\in\{1,2,\ldots,N_c\}} \frac{f(\mathbf{x}|c_i)\mathrm{p}(c_i)}{f(\mathbf{x})} . \tag{3}$$

Let $\mathbf{d}^j := \left\{\mathbf{d}_i^j\right\}$ denote the collection of all dyadic squares at scale j (each $\mathbf{d}^j$ contains complete information on the image $\mathbf{x}$). A posterior equivalent to (3) is then

$$\hat{c}_{i;j}^{\mathrm{MAP}} := \arg\max_{c_i^j\in\{1,2,\ldots,N_c\}} \frac{f(\mathbf{d}^j|c_i^j)\mathrm{p}(c_i^j)}{f(\mathbf{d}^j)} , \tag{4}$$

where $\frac{f(\mathbf{d}^j|c_i^j)p(c_i^j)}{f(\mathbf{d}^j)} = p(c_i^j|\mathbf{d}^j)$ is the marginal of the joint PDF $p(\mathbf{c}^j|\mathbf{d}^j)$ denoted as

$$p(\mathbf{c}^j|\mathbf{d}^j) = \frac{f(\mathbf{d}^j|\mathbf{c}^j)p(\mathbf{c}^j)}{f(\mathbf{d}^j)} = \frac{p(\mathbf{c}^j)}{f(\mathbf{d}^j)}\prod_i f(\mathbf{d}_i^j|c_i^j) \cdot \tag{5}$$

Equation (5) is based on the assumption that each $\mathbf{d}_i$ is generated with the distribution $f(\mathbf{d}_i|c_i)$ independently of all other class labels c_k and dyadic squares $\mathbf{d}_k$, $k \neq i$. However, the calculation of the marginal of $p(\mathbf{c}^j|\mathbf{d}^j)$ above is generally a difficult task unless $p(\mathbf{c}^j)$ has a special structure. In [2], a new organization structure, context labeling tree (CLT), was used to alleviate this difficulty. In this structure, the joint distribution of all the class labels C_i^j (the random variable corresponding to c_i^j) at scale j is completely determined by C_i^{j-1} at the immediately coarser scale and thus a Markov chain $\{C_i^{j-1}\} \to C_i^j \to \mathbf{d}_i^j$ is formed. Given $\mathbf{C}^{j-1} = \mathbf{c}^{j-1}$, the C_i^j s at scale j are independent, and the multiscale priori $p(\mathbf{c}^j|\mathbf{c}^{j-1}) = \prod_i p(c_i^j|\mathbf{c}^{j-1})$ holds. However, due to the high dimensionality of the conditioning vector $\mathbf{c}^{j-1}$, the estimate of the marginalized class priori distribution $p(c_i^j|\mathbf{c}^{j-1})$ still requires a prohibitive amount of training data [2]. In practice, a context vector $\mathbf{v}_i^j$, the function of the $\mathbf{c}^{j-1}$, is introduced to provide further simplification of the priori $p(\mathbf{c}^j|\mathbf{c}^{j-1})$. A notation $\mathbf{v}^j$ is used to denote the collection of all contexts at scale j. Conditioned on the context $\mathbf{v}^j$, equation (5) can be reformulated as

$$p(\mathbf{c}^j|\mathbf{d}^j,\mathbf{v}^j) = \frac{f(\mathbf{d}^j|\mathbf{c}^j)p(\mathbf{c}^j|\mathbf{v}^j)}{f(\mathbf{d}^j|\mathbf{v}^j)} = \frac{1}{f(\mathbf{d}^j|\mathbf{v}^j)}\prod_i \left[f(\mathbf{d}_i^j|c_i^j)p(c_i^j|\mathbf{v}_i^j)\right] \cdot \tag{6}$$

Thereby, the marginal $p(c_i^j|\mathbf{d}_i^j,\mathbf{v}_i^j) \propto f(\mathbf{d}_i^j|c_i^j)p(c_i^j|\mathbf{v}_i^j)$ is obtained, a simplified version of the MAP $p(c_i|\mathbf{x})$ in Eq. (3). The factor $f(\mathbf{d}_i^j|c_i^j)$ has been computed in the second step of HMTseg method, and the conditional probability $p\left(c_i^j|\mathbf{v}_i^j\right)$ can be denoted as

$$p\left(c_i^j|\mathbf{v}_i^j\right) = \frac{f(\mathbf{v}_i^j|c_i^j)p(c_i^j)}{\sum_{c_i^j=1}^{N_c} f(\mathbf{v}_i^j|c_i^j)p(c_i^j)} . \tag{7}$$

A new EM algorithm for CLT has been developed in [2] to solve for $f(\mathbf{v}_i^j|c_i^j)$ and $p(c_i^j)$. Thus far, the MAP estimate $\hat{c}_i^{\mathrm{MAP}}$ for the class label of each dyadic square $\mathbf{d}_i$ has been on hand.

3 Modified Image Segmentation Using Proposed Method

Effective modeling of context models for each dyadic square $\mathbf{d}_i$ is crucial to effectively fuse the ML raw segmentations from coarse scale to fine one in order to obtain good results in multiscale fusion stage. In the original HMTseg method [2], the context $\mathbf{v}_i^j$ is specified as a vector of two entries consisting of the value of class label $C_{\rho(i)}$ of the parent square and the majority vote of the class labels of the parent plus its eight neighbors, as illustrated in Fig. 2 (a). This simplified context is typically effective for images made up of separate large homogeneous textures since it lays more emphasis on the information of class labels at coarse scales. However, the segmentation results are mostly unsatisfactory when the images in hand, remotely sensed images for example, include more complex structures. Therefore, one slightly complicated and yet feasible context model, as shown in Fig. 2(b), is introduced here to incorporate both the information about the class labels at the coarse scale and that at the fine scale so as to take into account the region consistency and edge accuracy of segmentation performance simultaneously, which will be detailed in section 3.2.

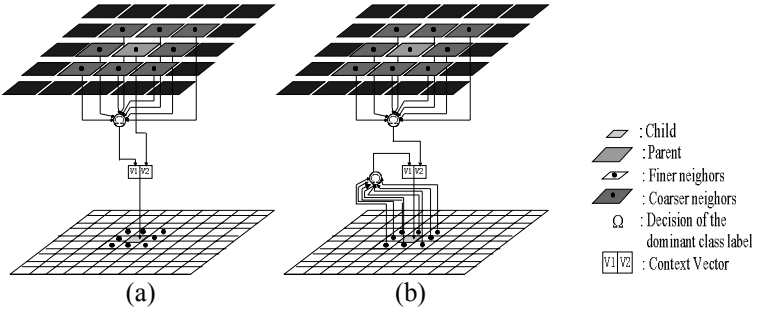


Fig. 2. (a) Context in [2] (b) new context proposed

3.1 Preprocessing of Raw Segmentations

As mentioned in section 2, the second step in the HMTseg method is multiscale likelihood computation by which the raw ML segmentations can be obtained. These raw segmentations are given only at the each individual scale without taking into the interactions across scales. Thus, the results from each single scale are typically unsatisfactory, which can be solved by the fusion of raw ML segmentations across scales using the third step in section 2. On the one hand, one appropriate context model is prerequisite to effectively implement fusion. On the other hand, modest preprocessing of raw segmentations could facilitate the following fusion and further improve segmentation performance. To this end, an 8-connectedness labeling [4] is introduced to amend the raw segmentations, which would favor better fusion results. This stage is carried out at each scale successively except for the coarsest scale of wavelet decomposition.

3.2 Multiscale Fusion Based on a New Context Model

Herein, a new context $\mathbf{v}_i^j$ for dyadic square $\mathbf{d}_i^j$ is defined, which consists of the majority vote of the class labels for the parent's eight neighbors and that for the child's eight neighbors.

The purpose of multiscale fusion is to maximize the conditional probability $p(c_i^j \left| \mathbf{d}_i^j, \mathbf{v}_i^j \right.)$ with which the label $\hat{c}_i^j$ for $\mathbf{d}_i^j$ can be found with MAP criterion, i.e.

$$\hat{c}_i^j = \underset{c\in\{1,2,\ldots,N_c\}}{\arg\max}\ p(c_i^j \left| \mathbf{d}_i^j, \mathbf{v}_i^j \right.) = \underset{c\in\{1,2,\ldots,N_c\}}{\arg\max}\ f(\mathbf{d}_i^j \left| c_i^j \right.) p(c_i^j \left| \mathbf{v}_i^j \right.), \tag{8}$$

where the computation of likelihood function $f(\mathbf{d}_i^j \left| c_i^j \right.)$ has been completed in raw segmentation stage. Hence, the key task is to calculate the conditional probability $p(c_i^j \left| \mathbf{v}_i^j \right.)$ based on which $\hat{c}_i^j$ can be obtained. According to Eq. (7), the probabilities $f(\mathbf{v}_i^j \left| c_i^j \right.)$ and $p(c_i^j)$ are undetermined temporarily. Here, we use the EM algorithm [2] specific for CLT to solve for $f(\mathbf{v}_i^j \left| c_i^j \right.)$ and $p(c_i^j)$. Similar to the settings in [2], $f(\mathbf{v}_i^j \left| c_i^j \right.)$ and $p(c_i^j)$ are assumed the same at each individual scale, and two notations $e_{j,n} := p(c_i^j = n)$, $\alpha_{j,\bar{v}_m,n} := p(\mathbf{v}_i^j = \bar{\mathbf{v}}_m \left| c_i^j = n \right.)$, $n \in \{1,\cdots,N_c\}$, $m \in \{1,\cdots,N_c^2\}$ are defined for all i at scale j. The EM algorithm to calculate $\mathbf{P} = \left\{ e_{j,n}, \alpha_{j,\bar{\mathbf{v}}_m,n} \right\}$ runs as follows.

Initialize: Set counter I=0 and choose appropriate P^0;

Expectation (E) step: Given $\mathbf{P}^I$, calculate

$$p(c_i^j = n \left| \mathbf{v}_i^j, \mathbf{d}_i^j \right.) = \frac{e_{j,n}\alpha_{j,\mathbf{v}_i^j,n} f(\mathbf{d}_i^j \left| c_i^j = n \right.)}{\sum_{c=1,\cdots,N_c} e_{j,c}\alpha_{j,\mathbf{v}_i^j,c} f(\mathbf{d}_i^j \left| c_i^j = c \right.)};$$

Maximization (M) step: Update the two elements of P^{I+1}

$$e_{j,n} = \frac{1}{2^{2j}} \sum_i p(c_i^j = n \left| \mathbf{v}_i^j, \mathbf{d}_i^j \right.),$$

$$\alpha_{j,\bar{v}_m,n} = \frac{1}{2^{2j} e_{j,n}} \sum_{i\ with\ \mathbf{v}_i^j = \bar{\mathbf{v}}_m} p(c_i^j = n \left| \mathbf{v}_i^j, \mathbf{d}_i^j \right.);$$

Iterate: Increment $I \to I+1$, and apply E and M steps until converged.

3.3 Pixel-Level Segmentation

Pixel-level segmentation can not be obtained directly from the ML raw segmentation procedure, since the wavelet transform (Haar wavelet base is adopted in this paper) characterizes the joint statistics of dyadic squares only down to 2×2 blocks [2]. As mentioned in section 2, the GMM can be exploited to model the pixel intensity values for each training texture, based on which the likelihood for each pixel is calculated and the multiscale fusion algorithm above can be naturally extended to the pixel level, and the final segmentation is accomplished. The segmentation of SAR images, however, can not be performed as well as natural textured images due to the intrinsic speckle in them. We use here the truncated HMTseg method [3] in which a scale threshold J was chose so that only the coefficients corresponding to dyadic squares not more than d^J were trained. Moreover, a combination strategy, using HMT-based raw segmentations and pixel-intensity-based ones, was utilized during inter-scale fusion procedure to finally get better results.

4 Experimental Results and Analysis

The experiments were conducted on remotely sensed images including an aerial photo (256×256 pixels, 256 gray levels) from USC-SIPC image database [5] and an SAR image (256×256 pixels, 256 gray levels, China Lake Airport, California, 3-m resolution) from Sandia national laboratories SAR image repository [6], shown in Fig. 3 (a) and (c). The HMT models for different types of textures (two classes for the aerial photo and three classes for the SAR image) were firstly trained based on the training data with size of 64×64 manually extracted from two original images. The number of wavelet decomposition levels was restricted to four with the balance between time consumption for model training and the reliability of segmentations in mind. The pixel-level segmentations were performed using GMM technique.

The final fusion results using original HMTseg method [2] and our method are demonstrated in Fig. 4. Not only is region consistency of segmentation performance obtained but the accuracy of boundary localization is improved further in Fig. 4 (b) and (d) obtained using the proposed method. For example, the ports in the aerial photo and the runways encircling the airports in the SAR image can be mostly discriminated. The results suggest that it is important to make full use of the information from coarse and fine scales simultaneously in the fusion process to gain better segmentation results.

5 Conclusion and Future Work

In this paper, a modified HMTseg method is proposed using a new context model and a preprocessing stage introduced to further amend the raw ML segmentations at different scales to facilitate the final multiscale fusion. Based on the proposed method, performance for remotely sensed images are improved, especially in the accuracy of boundary localization. Edges in an image provide important information

for identifying the objects in remotely sensed images. Combining the edge cues in an image to devise an edge-guided segmentation method is our further work.

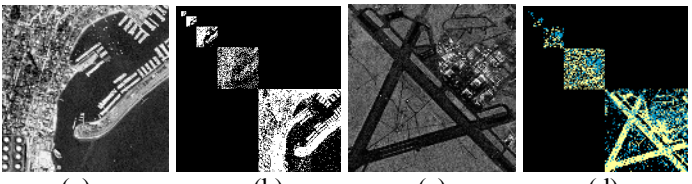


Fig. 3. Multiscale raw segmentation and pixel-level segmentation results of remotely sensed images: (a) aerial photo; (b) 4-level raw segmentation and pixel-level segmentation of (a); (c) SAR image; (d) 4-level raw segmentation and pixel-level segmentation of (c)

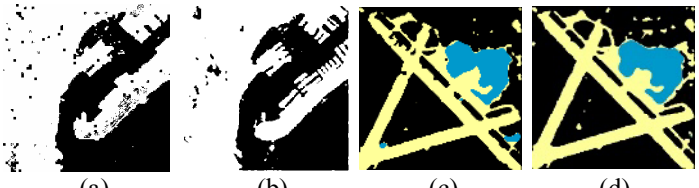


Fig. 4. Multiscale fusion results: (a) fusion result of Fig. 3 (b) by the method in [2]; (b) fusion result of Fig. 3 (b) by the proposed method; (c) fusion result of Fig. 3 (d) by the method in [2]; (d) fusion result of Fig. 3 (d) by the proposed method

References

1. Crouse, M.S. , Nowak, R.D. , Baraniuk, R.G.: Wavelet-Based Statistical Signal Processing Using Hidden Markov Models. IEEE Trans. on Signal Processing. 46 (1998) 886–902
2. Choi, H. , Baraniuk, R.G.: Multiscale Image Segmentation Using Wavelet-Domain Hidden Markov Models. IEEE Trans. on Image Processing. 10 (2001) 1309–1321
3. Venkatachalam, V. , Choi, H. , Baraniuk, R.G.: Multiscale SAR Image Segmentation Using Wavelet-Domain Hidden Markov Tree Models. In Proc. of SPIE, 4053 (2000) 1605–1611
4. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice-Hall, 1989
5. The USC-SIPC Image Database. Available: http://sipi.usc.edu/services.html
6. Sandia Synthetic Aperture Radar Imagery Repository. Available: http://www.sandia.gov/radar/imagery.html

Segmentation in Echocardiographic Sequences Using Shape-Based Snake Model

Chen Sheng[1], Yang Xin[1], Yao Liping[2], and Sun Kun[2]

[1] Institution of Image Processing and Pattern Recognition,
Shanghai Jiaotong University, Shanghai, P.R. China
chnshn@hotmail.com, yangxin@sjtu.edu.cn
[2] Shanghai Children's Medical Center, Shanghai Second Medical University,
Shanghai, P.R. China

Abstract. A novel method for segmentation of cardiac structures in temporal echocardiographic sequences based on the snake model is presented. The method is motivated by the observation that the structures of neighboring frames have consistent locations and shapes that aid in segmentation. To cooperate with the constraining information provided by the neighboring frames, we combine the template matching with the conventional snake model. Furthermore, in order to auto or semi-automatically segment the sequent images without manually drawing the initial contours in each image, generalized Hough transformation (GHT) is used to roughly estimate the initial contour by transforming the neighboring prior shape. As a result, the active contour can easily detect the desirable boundaries in ultrasound images.

1 Introduction

Endocardial boundary detection in ultrasound images is a necessary step to obtain both qualitative measurements (i.e., the detection of pathological deformation) and quantitative measurements (i.e., area, volume and etc.). Unfortunately, this is a difficult task due to the poor spatial and contrast resolutions, a high level of speckle noise and etc. To over come these problems, various algorithms are proposed to extract the boundaries of the region of interest (ROI) in echocardiographic images [1]. These approaches can be mainly categorized based on Markov random field [2], artificial neural network [3], mathematical morphological [4] and deformable model [5], etc. In these schemes, the deformable model [6], which is also known as the snake model, is the most important and popular model for noisy and low contrast image segmentation. In this paper, the main reason for using the snake model is that it allows the incorporation of geometric constraints.

However, the conventional deformable models have some deficiencies for boundary detection in ultrasound images. Firstly, the initial contour generally has to be placed quite close to the desirable boundary. Second, when the snake model is used to track the object in an image sequence by using the final contour from the previous frame as the initial contour in the current frame, the tracking works well only for small frame-to-frame displacement of ROI. Otherwise, the derived contour may be

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 375 – 382, 2005.

easily trapped in a local minimum formed by the noise. To remedy this problem, many techniques were proposed, for example, gradient vector flow (GVF) [7], dual snake [8] and discrete snake [9]. In this paper, it is noticed that the boundaries of any two adjacent images in a sequence are correlated to a certain degree. The result found in one image can be used as the shape template for the adjacent one. Thus, the only one rough shape template in a sequence needs to be given manually at the first step. For large frame-to-frame displacement of ROI, such as the mitral valve, GHT [10] is utilized to transform the shape template to an initial contour in the ROI. It has been proven that GHT is able to detect any arbitrary shape undergoing a geometric transformation in an image. Moreover, it has shown to be robust and can even be successfully used to detect overlapping or semi-occluded objects in noisy images.

Our method is based on the template matching which incorporates the prior shape template from the adjacent frame into the snake model. Optimizing the deformation energy between the shape template and the active contour, the shape of the active contour is constrained to be similar to the template in global while still allowing slight deformation locally. Under this energy criterion, the contour can escape from the local minimum caused by the speckle, the tissue-related textures.

2 Methods

Let Ω be a bounded open subset of R^2. Let $u_0 : \overline{\Omega} \to R$ be a given image, and $C(s) = (x(s), y(s))(s \in [0,1])$ be a parameterized contour with s the parameter of length. The shape-based snake model is to minimize the following energy:

$$E(u_0, C_d, C_t) = \alpha E_{\text{int}}(C_d) + \beta E_{ext}(u_0, C_d) + \eta E_{con}(C_d, C_t) \tag{1}$$

where C_d is the active contour, C_t is the shape template.

$E_{\text{int}}(C_d)$ is the internal energy that controls the smoothness of the contour:

$$E_{\text{int}}(C_d) = \int_0^1 \left| C'_d(s) \right|^2 ds + \int_0^1 \left| C''_d(s) \right| ds \tag{2}$$

$E_{ext}(u_0, C_d)$ is the external energy that attracts of the active contour evolving to the boundary of object. In this paper, it is calculated from the texture information instead of the local gradient in the ultrasound image. Furthermore, a blurring Gaussian filter is applied for better result. However, the blurred texture feature probably loses some object boundary information. Hence, the original image feature is also used to retain the boundary information. Let $T(x, y)$ denote the texture image after applying the texture analysis to the original image $u_0(x, y)$. The blurring Gaussian filter is applied to the texture image $T(x, y)$ to obtain the blurred texture image $T_G(x, y)$. Now the external energy E_{ext} is defined as:

$$E_{ext}(u_0, C_d) = -\left| \nabla u_0(C_d(s)) \right| - \left| \nabla T_G(C_d(s)) \right| \tag{3}$$

$E_{con}(C_d, C_t)$ is the energy to measure the similarity between the active contour and the shape template. In this paper, our method has been inspired by the approach due to Duncan [11], who proposed a scheme for matching two contours based on the minimization of a quadratic fitting criterion, which consists of a curvature dependent bending energy term and a smoothness term.

Duncan [11] introduces a local bending energy measure of the form:

$$E_{curvature} = \int \left(k_{C_d}(s') - k_{C_t}(s)\right)^2 ds \tag{4}$$

where $k_{C_d}(s')$ is the curvature of the active contour C_d at s' as well as $k_{C_t}(s)$.

We also wish the displacement vector field to vary smoothly along the active contour:

$$E_{smooth} = \int \left\| \frac{\partial \left(C_d(s') - C_t(s)\right)}{\partial s} \right\| ds \tag{5}$$

So the criterion is composed of the curvature constraint and the smooth constraint:

$$E_{elastic} = E_{curvature} + \lambda E_{smooth} \tag{6}$$

3 Initialization of the Active Contour

The shape template must be approximated as a vector containing a sequential discrete points in order to solve by numerical method, $U = [u_1, u_2, \cdots, u_n]$, where $u_i = (u_{ix}, u_{iy}) \in \{(x, y): x, y = 1,2,\cdots\cdots, M\}$. The same method is used for the active contour, $V = [v_1, v_2, \cdots, v_n]$.

Before processing the boundary detection by the snake, an initial contour must be draw. The purpose of the initialization is to place the initial contour as close as possible to the boundary in ROI in order to obtain a fast convergence in the boundary detection. In this paper, the GHT is applied to solve this problem. Let us define a geometric transformation of the shape template by:

$$V = AU + t = \begin{bmatrix} a_A & b_A \\ c_A & d_A \end{bmatrix} \cdot \begin{bmatrix} U_x \\ U_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \qquad (a_A d_A - b_A c_A \neq 0) \tag{8}$$

where A and t correspond to a linear transformation and to a translation vector respectively. The potential location of the position parameters t for the potential parameters A of the linear transformation can be expressed as $t(U, V, A) = V - AU$. This method traces an initial contour in the parameter space, and after gathering all evidences for all ROI pixels, the maximum of the accumulator array defines the best values A^* and t^* which correspond to the transformation that maps the shape template to the echocardiographic image. The GHT can deliver a reliable estimation of the ROI position or a coarse initial contour.

4 Experiments and Results

In this section, several examples are presented to illustrate the efficiency of the shape-based snake model for boundary detection in echocardiographic sequences. Six sequent ultrasound images with size 180×180 pixels were obtained from the Philip 5500 system, each covering one complete cardiac cycle and containing $F=16$ frames. The algorithm has been implemented using an Intel Pentium IV 2.4Ghz with 1 GB RAM, under the Visual C++ 6.0 environment.

To assess the performance of our segmentation method, we compared automatically detected cardio structure boundaries with the manual outlines. In this paper, four sets of manual outlines are given for each of the sequences.

Two sets of parameters are employed: the mean, the standard deviation (SD), and the maximum of the minimal distances from the derived boundary points to the manual outline. They are used to measure the difference between the derived contour and the outline in one frame of a sequence.

Let C_d and C_m denote the derived contour and the manual outline, respectively.

~ For each $p_i \in C_d$, find $p_i^* \in C_m$ so that $p_i^* = \arg\min_{\forall p_j^* \in C_m} \left\| p_i - p_j^* \right\|$, where $\left\| p_i - p_j^* \right\|$ means the Euclidean distance between the two pixels.

~ For all (p_i, p_i^*), computer the Euclidean distance d.

~ Compute the mean, the SD and the maximum of $\left\{ d \middle| \forall p_i \in C_d \right\}$.

We need another set of parameters to evaluate the segmentation results for the whole sequence, so the mean and the SD of the mean absolute distance (MAD) are defined as follow:

~ The MAD between two contours A and B is defined as: $D(A,B) = \frac{1}{2}\left\{ \frac{1}{n}\sum_{i=1}^{n} d(a_i, B) + \frac{1}{m}\sum_{i=1}^{m} d(b_i, A) \right\}$.

~ Compute the mean, the SD of $\left\{ D \middle| \forall D \in S \right\}$, where S is all the MADs need to be calculated for a sequence.

4.1 Process of Segmentation

Fig.1 shows the segmentation process for a mitral valve sequence. The initial contour of the $(k+1)$ th image obtained directly from the final contour of the k th image is shown in Fig.1 (c). Fig.1 (b) presents the initial contour, which has been transformed by GHT. In the Fig.1 (d), we can see that the segmentation result rather coincides with the contour manually defined by an independent doctor in Fig.1 (e) when using GHT to locate the initial contour. On the other hand, we can see that the shape-based snake model treats well when there is a gap in the tip of the leaflet under the shape constraint.

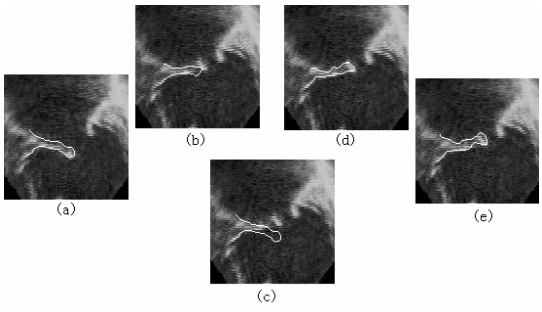


Fig. 1. Example of segmentation for mitral valve; (a) the kth image with finial contour; (b) the (k+1)th image with initial contour from the kth image using GHT; (c) the (k+1)th image with initial contour direct from the kth image; (d) the (k+1)th image with segmentation result using initial contour in (b); (e) manual outline for the (k+1)th image

Table 1. The mean, the SD and the maximum of the minimal distances for Fig.2

Minimal distances	Mean [pixels]	SD [pixels]	Max [pixels]
Using GHT	1.5	1.06	5.2
Without using GHT	1.4	1.15	5.1

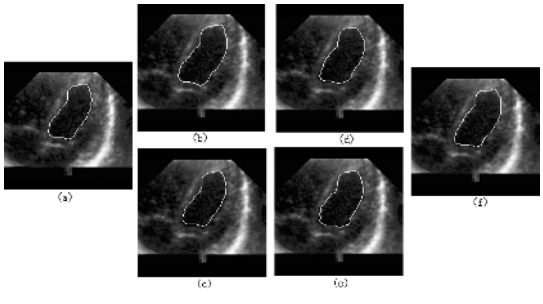


Fig. 2. Example of segmentation for left ventricle; (a) the kth image with finial contour; (b) the (k+1)th image with initial contour from the kth image using GHT; (c) the (k+1)th image with initial contour direct from the kth image; (d) the (k+1)th image with segmentation result using initial contour in (b); (e) the (k+1)th image with segmentation result using initial contour in (c); (f) manual outline for the (k+1)th image

It may be reasonable to say that the segmentation result closely follows the desired boundary. Nevertheless, the algorithm fails when using the initial contour in Fig.1 (c) although the same energy weighting factors ($\alpha = 1.0, \beta = 1.0, \eta = 0.5$) are given.

GHT is not needed in all situations such as the small frame-to-frame displacement of the structure. Fig.3 (e) shows the segmentation result for the left ventricle with the initial contour direct from the previous image is identical to that using GHT to locate the initial contour (Fig.3 (d)). The evaluated parameters of the segmentation results are shown in Table 1. Both the mean and the SD of the minimal distances are near to each other.

4.2 Segmentation of Endocaridal Boundaries in Sequences

In four sequences, the algorithm was used to segment the endocaridal boundaries. Some frames from the first sequence are shown in Fig.3.

Table 2 shows the mean and the SD of the MADs for the whole sequence between the algorithm-generated contours and the four sets of manual outlines and between different manual outlines. These experiments show that the segmentation results compare well to the manual outlines for the endocaridal boundaries.

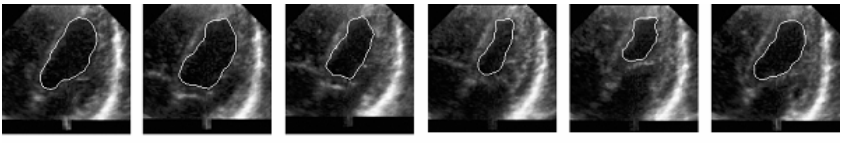


Fig. 3. Characteristic frames showing the segmentation results of the left ventricle

Table 2. Results of the comparison between the algorithm-generated contours and the manual outlines

	Seq1	Seq2	Seq3	Seq4
Mean of MADs between snake and outlines [pixels]	1.22	1.75	1.61	1.18
SD of MADs between snake and outlines [pixels]	0.26	0.35	0.45	0.31
Mean of MADs between different manual outlines [pixels]	1.32	1.65	1.41	1.24
SD of MADs between different manual outlines [pixels]	0.22	0.34	0.30	0.28

4.3 Segmentation of Mitral Valve Sequences

The algorithm performance was evaluated on two sequences of long axis view images of the mitral valve. Characteristic frames from the first sequence are shown in Fig.4. As one could expect, the differences of ROI between any two adjacent frames are larger, but the algorithm performance is still comparable to the manual segmentations.

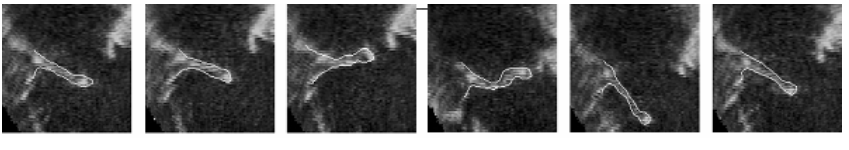


Fig. 4. Characteristic frames showing the segmentation results of the mitral valve

Table 3. Results of the comparison between the algorithm-generated contours and the manual outlines for sequences containing images of the mitral valve

	Seq1	Seq2
Mean of MADs between snake and outlines [pixels]	2.14	2.02
SD of MADs between snake and outlines [pixels]	0.71	0.56
Mean of MADs between different manual outlines [pixels]	1.81	1.63
SD of MADs between different manual outlines [pixels]	0.69	0.52

Table 3 shows the evaluated results for mitral valve sequences. In this Table, we can see that the mean and the SD are larger than those in Table 2. It may be ascribed to at least two factors. The first one is that the manual outlines may vary with experts. The second factor is that the contours in the mitral valve images are open. The starting points and the ending points defined by the experts may vary largely. As a result, the MAD between the open contours may be larger than that between closed contours.

4.4 Determination of Weighting Factors

In our experiments, the weighting factors α, β, η are set in Table 4. The motion of the mitral valve is very irregular, frame-to-frame displacements are several times larger than the leaflet thickness. At those phases, the leaflet rotates, translates and deforms at the same time. As a result, the difference of shape between two adjacent images may be large.

Table 4. Values of parameters used in the algorithm

	α	β	η
endocaridal sequences	1.0	1.0	2.0
Mitral valve sequences	1.0	1.0	0.5

GHT algorithms are known to be computationally expensive (about 6 min for a sequence in our experiments) and they are not needed in all situations. So, in our method, the GHT was separated from the snake deformation process. A user can intervene when or where GHT to be used. However, these algorithms do not need user's supervision during the segmentation process. The user's interaction was needed in just one frame for a sequence.

5 Conclusions

In this paper, an innovative model has been proposed for echocardiographic image segmentation, namely, the shape-based snake model. The proposed shape-based model aims to incorporate the template matching and the GHT with the snake model. The model can resist the speckle noise, tissue-related textures and artefacts, and guide the active contour deform to the desirable boundary. The principal idea of this model is to use GHT to estimate the initial contour, and then using the elastic deformation energy between the shape template and the active contour to guide the contour deform from the local minimum. Our method does not need to draw a precise shape template, but rather a rough contour regardless of its position, scaling and rotation only once in a sequence.

Acknowledgments

This work was partially supported by National Science Research Program of China (No. 2004BA714802) and Shanghai Science and Technology Development Foundation (034119820).

References

1. Sher DB, Revankar S. Computer methods in quantitation of cardiac wall parameters from two-dimensional echocardiograms: A survey. Int. J. Cardiac Imaging 1992;8;11-26.
2. Dias JMB, Leitao JMN. Wall position and thickness estimation from sequence of echocardiographic images. IEEE Trans on Medical Imaging 1996;15;25-38.
3. Kotropolulos C. Nonlinear ultrasonic image processing based on signak-adaptive filters and self-organizing neural networks. IEEE Trans Image Processing 1994;3;65-77.
4. Thomas JG, Peters RA, Jeanty P. Automatic segmentation of ultrasound images using morphological operators. IEEE Trans Medical Imaging 1991;10;180-186.
5. Hass C, Ermert H, Holt S. Segmentation of 3-D intravascular ultrasonic images based on a random field model. Ultrasound Med Biol 2000;26;297-306.
6. Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. International Journal of Computer Vision 1987;1;321-331.
7. Xu C, Prince JL. Snakes, shapes, and gradient vector flow. IEEE Trans Image Processing 1998;7;359-369.
8. Gunn SR, Nixon MS. A robust snake implementation: A dual active contour. IEEE Trans PAMI 1997;19;63-68.
9. Lobregt S, Viergever MA. A discrete active contour model. IEEE Trans Medical Imaging 1995;14;12-24.
10. 10.Ballard DH. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition 1981;13;111-122.
11. 11.Duncan JS, Owen R. Shape-based tracking of left ventricular wall motion. Computers in Cardiology 1990. IEEE Computer Society, Chicago, Illinois. 1990 September;23-26.

An Algorithm for Binary Image Segmentation Using Polygonal Markov Fields

Rafał Kluszczyński[1], Marie-Colette van Lieshout[2], and Tomasz Schreiber[1]

[1] Nicolaus Copernicus University, ul. Chopina 12-18, 87-100 Toruń, Poland
klusi@mat.uni.torun.pl, tomeks@mat.uni.torun.pl
[2] CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
Marie-Colette.van.Lieshout@cwi.nl

Abstract. We present a novel algorithm for binary image segmentation based on polygonal Markov fields. We recall and adapt the dynamic representation of these fields, and formulate image segmentation as a statistical estimation problem for a Gibbsian modification of an underlying polygonal Markov field. We discuss briefly the choice of Hamiltonian, and develop Monte Carlo techniques for finding the optimal partition of the image. The approach is illustrated by a range of examples.

1 Introduction

One of the fundamental image analysis tasks is that of *segmentation*, i.e. to partition the image in relatively homogeneous regions [10]. Indeed, segmenting the data is often the first step in image interpretation problems. The partition may be achieved at several conceptual levels. At the lowest level, that of individual pixels, segmentation amounts to classification of pixel values. At the other extreme, the focus of attention are the objects that constitute a given image and the goal is to extract them from the image.

A myriad of segmentation methods has been proposed, from elementary thresholding through level set approaches and contour extraction methods to scene modelling. In this paper, we propose to use polygonal field models. Thus, we place ourselves at the intermediate conceptual level that regards a segmentation as a coloured tessellation [8]. The advantage of such an approach is that - in contrast to pixel based ones - global aspects of the image are captured. At the same time there is no need to model all objects in the image, which is feasible in restricted application domains only. Furthermore, a coloured polygonal tessellation is a reasonable and widely applicable mathematical formalisation of the intuitive concept of a segmentation, especially when compared to the rather artificial level set model or to the notion of a collection of pixel labels that do not necessarily have any spatial coherence. The idea can be traced back to Clifford and Middleton [6]; from a computational point of view, a Metropolis–Hastings style sampler was developed by Clifford and Nicholls [7] and applied to an image reconstruction problem within a Bayesian framework. We shall use a modification of the algorithm in Schreiber [12] which is conceptually and computationally easier.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 383–390, 2005.

2 Preliminaries and Notation

We aim to describe the contents of an image domain D, assumed to be an open, convex and bounded subset of the plane (typically a rectangle), by means of a family of non-intersecting polygonal contours in D, possibly nested or chopped off by the boundary. We restrict ourselves to foreground/background segmentation, so that a contour may be interpreted as a polygonal boundary between black (foreground) and white (background) regions. We shall use the notation $\bar{y}$ for a discretised image, $S \subset D$ for the pixel set. The value y_s at pixel s belongs to some finite set L. A collection of contours is denoted by γ. For each collection, there are exactly two admissible black and white colourings. We use a hat notation, $\hat{\gamma}$, to describe a family of non-intersecting polygonal contours with its associated colouring.

Our approach, as developed in [12] and [9], involves the following building blocks.

1. The first ingredient is, intuitively speaking, the ability to generate a 'completely random' polygonal field, both as a benchmark or reference field, and as a tool for exploring the space of admissible polygonal configurations without favouring any particular one. A reasonable choice is the so-called Arak process [1,2,3,4], denoted by $\mathcal{A}_D$.
2. Secondly, we need a *goodness-of-fit measure* $\mathcal{H}(\bar{y} \mid \hat{\gamma})$ to quantify how well a coloured polygonal configuration $\hat{\gamma}$ matches the data image $\bar{y}$. Moreover, we would like to be able to influence the geometry of $\hat{\gamma}$ by assigning a higher probability to large polygons with smooth boundaries that do not have spurious edges. This is captured by a so-called *regularisation* term $\mathcal{H}(\bar{y})$, cf. Section 3 in [9].
3. The third and last ingredient is an updating mechanism which keeps the distribution of the reference Arak field invariant while exhibiting good exploratory properties in the configuration space. To this end, we develop the so-called disagreement-loop birth and death algorithm, originally introduced in [12] and further extended in [9]. This mechanism can then be combined with standard Metropolis and simulated annealing techniques to find an optimal segmentation $\hat{\gamma}$.

The next three sections will elaborate a little further on the above points. A full account can be found in [12,9].

3 The Arak Process: Dynamic Representation

The crucial idea underlying Arak's construction [1] is to interpret the polygonal boundaries of the field as the trace left by a particle travelling in two-dimensional time-space. Thus, the two-dimensional image domain D is seen as a set of *time-space* points $(t, y) \in D$, with t referred to as the *time* coordinate and with y standing for the (1D) *spatial* coordinate. In this language, the basic Arak process is constructed as follows.

Birth events. The new particles are born in pairs at *birth sites* chosen in the interior of the domain D according to a homogeneous Poisson point process of intensity π. There are also *boundary birth sites* emitting single particles, generated by a Poisson point process with an appropriate intensity measure $\kappa(\cdot)$ concentrated on ∂D, whose analytic details we omit here for simplicity of presentation, referring the reader to [2] for an exhaustive description.

* Each interior birth site emits two particles, moving with initial velocities v' and v'' chosen according to the joint distribution $\theta(dv', dv'') := \pi^{-1}|v' - v''|(1+v'^2)^{-3/2}(1+v''^2)^{-3/2}dv'dv''$ on $v' < v''$. This is equivalent to choosing the angle $\phi \in (0, \pi)$ between the straight lines representing the space-time trajectories of the emitted particles according to the density $\sin(\phi)/2$.
* Each boundary birth site $x \in \partial D$ yields one particle with initial speed v determined according to an appropriate distribution $\theta_x(dv)$, see [2] for its explicit form.

The colour in the interior of the newly created angle is chosen so as not to clash with the one to the left of the trajectory (the past in time-space terms), with minor modification for left-extreme points.

Evolution rules. All the particles evolve independently in time according to the following rules.

* Between the critical moments listed below each particle moves freely with constant velocity, hence $dy = vdt$.
* When a particle touches the boundary ∂D, it dies.
* In case of a collision of two particles (equal spatial coordinates y at some moment t with $(t, y) \in D$), both of them die.
* The time evolution of the velocity v_t of an individual particle is given by a pure-jump Markov process so that $\mathbf{P}(v_{t+dt} \in du \mid v_t = v) = q(v, du)dt$ for the transition kernel $q(v, du) := |u - v|(1 + u^2)^{-3/2}du$.

The random polygonal configuration obtained in the above procedure is precisely the basic Arak process in D. As shown in [2], $\mathcal{A}_D$ enjoys a number of striking properties. One of them is the two-dimensional germ *Markov property*, stating that the conditional distribution of the field inside an open bounded region with piecewise smooth boundary given the outside configuration depends only on the trace of this configuration on the boundary (colouring, intersection points and intersection directions). The next important property is *consistency*: for bounded open and convex D_1 and D_2 with $D_1 \subseteq D_2$ the restriction of $\mathcal{A}_{D_2}$ to D_1 coincides in distribution with $\mathcal{A}_{D_1}$, see Theorem 4.1 of [2]. A crucial property is the isometry invariance of the Arak process - while the translational invariance can be easily deduced from the construction above, the rotational invariance is a deep and non-trivial result that follows from the particular choice of the kernels $\theta.(\cdot)$ and $q(\cdot, \cdot)$ above. Moreover, one dimensional sections of $\mathcal{A}_D$ happen to be homogeneous Poisson point processes. Finally, a number of explicit formulae are available for various numeric characteristics of $\mathcal{A}_D$, such as the expected total edge length, mean number of vertices, edges etc, see [2,4,7]. These properties suggest that the Arak process is a suitable reference field.

4 Model-Based Image Segmentation

The model to be used for inference will be a Gibbsian modification of the polygonal random field $\mathcal{A}_D$ by means of a Hamiltonian (sometimes referred to as energy function)

$$\mathcal{H}(\bar{y};\hat{\gamma}) = \mathcal{H}(\bar{y} \mid \hat{\gamma}) + \mathcal{H}(\hat{\gamma}) \tag{1}$$

that is the sum of two terms, cf. Section 2. In other words, upon observation of $\bar{y}$, the likelihood of $\hat{\gamma}$ with respect to the reference field is weighted by a factor $\exp[-\mathcal{H}(\bar{y};\hat{\gamma})]$, then normalised to have total probability mass 1. We take $\mathcal{H}(\bar{y} \mid \hat{\gamma}) = L_1(\bar{y},\hat{\gamma})$, the sum of absolute differences between the actual pixel values and those assigned by $\hat{\gamma}$. For binary images, the L_1 distance reduces to $|\bar{y}\Delta\hat{\gamma}|$, the cardinality of the set of sites at which the observed colour does not match that of the polygon, which can be interpreted probabilistically as a random noise model in which each pixel value is flipped to the wrong colour independently of other pixels with some probability $p < 1/2$ (see [5,9] for details). Thus, finding an optimal $\hat{\gamma}^*$ in the absence of further regularisation would amount to minimising the misclassification rate $|\bar{y}\Delta\hat{\gamma}|/|S|$. In general, $\hat{\gamma}^*$ is not unique. Moreover, it tends to result in an over-segmentation. To overcome these problems, we added a regularisation term $\mathcal{H}(\hat{\gamma}) = \beta\,\ell(\gamma)$ proportional to the total edge length.

We are now ready to rephrase image segmentation as the task of finding a coloured polygonal configuration $\hat{\gamma}$ and a real β so that the Hamiltonian value in (1) is a sufficiently good approximation of the global minimum.

5 Disagreement Loop Birth and Death Dynamics

To minimise (1), we use simulated annealing. Briefly, given a sequence of temperatures T_n decreasing to zero as $n \to \infty$, we sample from a probability density proportional to $\exp[-\mathcal{H}(\bar{y};\hat{\gamma})/T_n]$ with respect to the reference field. Clearly it suffices to describe how to sample for $T_n = 1$. A crucial concept in our algorithm is that of a *disagreement loop* [11, Section 2.2]. Consider adding a new birth site x_0 to configuration γ, and denote the resulting polygonal configuration by $\gamma \oplus x_0$. Then, for $x_0 \in \mathrm{Int}\ D$, the symmetric difference $\Delta^{\oplus}[x_0;\gamma] := \gamma \triangle [\gamma \oplus x_0]$ is just a single loop (a closed polygonal curve), possibly self-intersecting and possibly chopped off by the boundary. Likewise, a disagreement loop $\Delta^{\ominus}[x_0;\gamma]$ arises by removing a birth site x_0 in polygonal configuration γ; we write $\gamma \ominus x_0$ for the resulting configuration. This is discussed in details in [11,12], here we only give an illustration in Fig. 1. Recall that $\kappa(\cdot)$ is the boundary birth intensity measure as in Section 3. Then,

(DL:birth) with intensity $[\pi dx + \kappa(dx)]ds$ set $\delta := \gamma_s \triangle \Delta^{\oplus}[x;\gamma_s] = \gamma_s \oplus x$. Choose either of the two admissible colourings with probability 1/2 to obtain $\hat{\delta}$. Then, with probability $\min\left(1, \exp\left[\mathcal{H}(\bar{y};\hat{\gamma}_s) - \mathcal{H}(\bar{y};\hat{\delta})\right]\right)$ put $\hat{\gamma}_{s+ds} := \hat{\delta}$, otherwise keep $\hat{\gamma}_{s+ds} := \hat{\gamma}_s$;

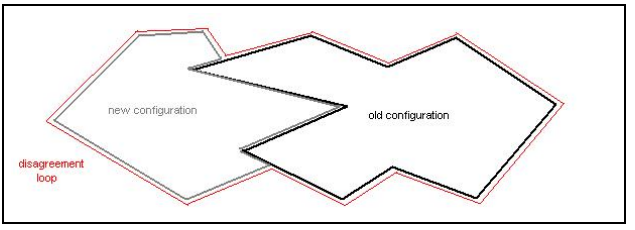


Fig. 1. Disagreement loop

(DL:death) for each birth site x in γ_s, with intensity $1 \cdot ds$ set $\delta := \gamma_s \triangle \Delta^{\ominus}[x; \gamma_s] = \gamma_s \ominus x$. Choose either of the two admissible colourings with probability $1/2$ to obtain $\hat{\delta}$. Then, with probability $\min(1, \exp[\mathcal{H}(\bar{y}; \hat{\gamma}_s) - \mathcal{H}(\bar{y}; \delta))])$ put $\hat{\gamma}_{s+ds} := \hat{\delta}$, otherwise keep $\hat{\gamma}_{s+ds} := \hat{\gamma}_s$.

By Theorem 1 in [12] and Theorems 1–2 in [9], under these dynamics the current polygonal configuration converges in distribution to the Gibbsian modification of $\mathcal{A}_D$ with the Hamiltonian $\mathcal{H}(\bar{y}; \hat{\gamma})$. This algorithm exploits the fact that disagreement loops $\Delta^{\oplus,\ominus}[x; \gamma]$ are very well suited for simulation. Since both the dynamic representation of the Arak process in Section 3 and the simulation algorithm presented here involve a notion of time, in the sequel we refer to the former as to the *representation time* (r-time) and to the latter as to the *simulation time* (s-time).

Extensions of the algorithm. In order to increase the efficiency of the basic algorithm we endow it with a number of extra Monte Carlo moves. These include

1. Random rotations of the spatial and time axes in the dynamic representation of the Arak process. While enlarging the set of possible moves, this does not alter the stationary distribution of the process due to the isometry invariance of $\mathcal{A}_D$.
2. Repetitive velocity updates: a particle undergoing a velocity update in r-time, is allowed to 'change its mind' in the course of s-time and to randomly modify the previously performed update. This can also be done so that the stationary distribution is unaltered.
3. Rescaling, to guarantee better resolution at later stages of annealing.

6 Implementation and Examples

The algorithm was implemented in `C++`. General features are discussed briefly below.

Representation of polygonal configurations. A configuration of a polygonal Markov field is represented as a list of labelled vertices. The full description of a vertex is provided by

– the Cartesian coordinates of the vertex;
– two pointers to the neighbouring vertices;

– the *virtual lengths* of the segments that emanate from the given vertex; these are the lengths the segments would have if the corresponding particles were the only ones present in the system and evolved in an empty environment. The actual lengths of these segments are usually smaller due to collisions.

The list of vertices is sorted by increasing x-coordinate (r-time coordinate).

Generation of the initial configuration. The initial configuration for our MCMC procedure is generated according to the dynamic representation of the basic Arak process as discussed in Section 3. This is done in a single left-to-right sweep through the image domain, by successively updating in r-time two priority queues that store respectively

– the birth sites with r-time coordinate exceeding the current r-time, and *virtual end points* (with the distance from the respective initial point given by the corresponding virtual length) of segments generated so far, for which the r-time coordinate exceeds the current r-time;
– *virtual collision points* which are all possible pairwise intersection points of currently existing *virtual segments* (i.e. segments joining an initial point to its corresponding virtual end point) lying forward in r-time.

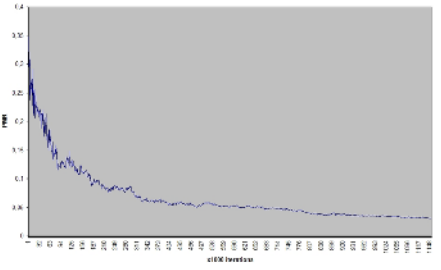
At each step of the algorithm, the next vertex to arise in the course of the r-time evolution is determined by choosing that vertex that minimises the r-time coordinate in both queues. Consequently, the contents of these queues can be regarded as a current collection of 'candidates' for the next vertex.

Updates. Our main configuration-modifying operations are adding a new birth site (disagreement loop birth **(DL:birth)**) and removing an existing one (disagreement loop death **(DL:death)**). To add a new birth site, we first choose its position uniformly at random within the image domain, and then we let the newborn particles evolve and interact with the existing ones according to the usual evolution rules. Likewise, when removing a given birth site, we let the remaining particles obey the usual evolution rules. Both these updates are implemented using the same data structures as when generating the initial configuration above.

Evaluation of the Hamiltonian. For binary images $\bar{y}$, $\mathcal{H}(\bar{y} \mid \hat{\gamma})$ requires the evaluation of the number of misclassified pixels upon each update proposal. To this end we apply the divergence theorem, constructing a real-valued vector field such that the input image data $\bar{y}$ coincides with its divergence, and then computing appropriate flux integrals along the suitably oriented contours of the polygonal field. For grey level images, we resort to Monte Carlo sampling to calculate the L_1 distance between $\bar{y}$ and $\hat{\gamma}$.

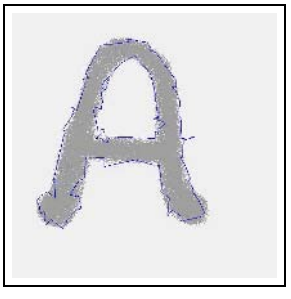
Examples. Here we present a few typical segmentations obtained by the approach. The data in the first example consist of a spray-traced image of a happy face. For the cooling schedule we used $1/T_n := 20.0 + 0.009 * n$. The result after approximately 1 150 000 steps is given in Figure 2a. The misclassification rate we achieved was 3 percent, as visualised in the corresponding graph [Fig.

(a) segmented 'happy face'

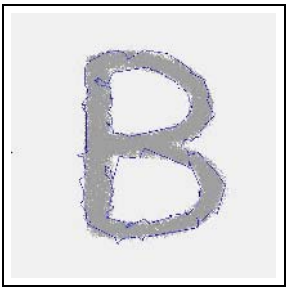


(b) misclassification rate graph

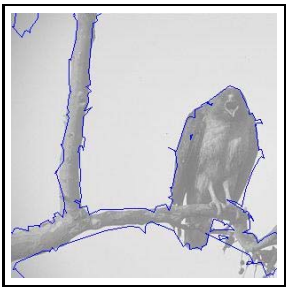
Fig. 2



(a) segmented letter A



(b) segmented letter B



(c) segmented eagle

Fig. 3

2b]. Average execution time for a single iterate was 0.0061 second for 2 CPUs architecture with Intel Xeon 2.4 GHz processors and 2 GB RAM. The data of the second and third image are given by the first letters of the alphabet, 'A' and 'B'. The results under the same cooling regime are given in Figures 3a and 3b, and were achieved after 800 000 and 2 million iterations respectively. The misclassification rate is 3 percent in both cases. Finally, Fig. 3c presents a sample segmentation of a grey level eagle image from the Berkeley segmentation dataset and benchmark site. The misclassification rate reached after 860 000 iterations is again 3 percent.

7 Discussion and Future Work

Here, for brevity, we restricted ourselves to foreground/background segmentation. However, the general framework can easily be extended to allow for segmentation into $k > 2$ classes of grey level, colour, or textured images [9]. Suitable consistent polygonal field models are available with more flexibility in the type of intersections [4], but care is needed with respect to the Hamiltonian. Indeed, it is the object of our current work to develop software for such more complicated cases, and to evaluate the performance of our algorithm on benchmark data available at: `http://www.cs.berkeley.edu/projects/vision/grouping/segbench/` or `http://mosaic.utia.cas.cz`.

A preliminary conclusion is that, in contrast to model-based probabilistic image segmentation at the pixel level, the topology of the foreground object is preserved better. Our approach is also relatively robust with respect to noise. The price to pay is that fine details are not recovered, especially those whose sizes fall below the characteristic scale of the polygonal field. This problem could easily be solved in a post-processing smoothing phase. An alternative could be to gradually decrease the characteristic scale of the field (multi-resolution approach) or to build local updates in the spirit of [7] into the algorithm.

Acknowledgements

This research was supported by the EC 6th Framework Programme Priority 2 Information Society Technology Network of Excellence MUSCLE (Multimedia Understanding through Semantics, Computation and Learning; FP6-507752) and by the Foundation for Polish Science (FNP) and the Polish Minister of Scientific Research and Information Technology grant 1 P03A 018 28 (2005-2007).

References

1. Arak, T. (1982) On Markovian random fields with finite number of values. In *4th USSR-Japan Symposium on Probability Theory and Mathematical Statistics, Abstracts of Communications*, Tbilisi.
2. Arak, T. and Surgailis, D. (1989) Markov fields with polygonal realisations. *Probability Theory and Related Fields* **80**, 543–579.
3. Arak, T. and Surgailis, D. (1991) Consistent polygonal fields. *Probability Theory and Related Fields* **89**, 319–346.
4. Arak, T., Clifford, P. and Surgailis, D. (1993) Point-based polygonal models for random graphs. *Advances in Applied Probability* **25**, 348–372.
5. Baddeley, A.J. and van Lieshout, M.N.M. (1992) ICM for object recognition. In *Computational Statistics*, Y. Dodge and J. Whittaker (Eds.), volume 2, pp. 271–286. Physica/Springer, Heidelberg.
6. Clifford, P. and Middleton, R.D. (1989) Reconstruction of polygonal images. *Journal of Applied Statistics* **16**, 409–422.
7. Clifford, P. and Nicholls, G.K. (1994) A Metropolis sampler for polygonal image reconstruction. Electronic version available at: `http://www.stats.ox.ac.uk/~clifford/papers/met_poly.html`.
8. Hurn, M.A., Husby, O. and Rue, H. (2003) Advances in Bayesian image analysis. In *Highly Structured Stochastic Systems*, P.J. Green, S. Richardson and N.L. Hjort (Eds.), Oxford Statistical Science Series **27**, 323–325. Oxford University Press, Oxford.
9. Kluszczyński, R., van Lieshout, M.N.M. and Schreiber, T. (2004) Image segmentation by polygonal Markov fields. Electronic version available as CWI Research Report PNA-R0409 at: `http://www.cwi.nl/publications`.
10. Rosenfeld, A. and Kak, A.C. (1982) *Digital picture processing*, second edition, volume 2. Academic Press, Orlando.
11. Schreiber, T. (2003) Mixing properties of polygonal Markov fields in the plane. Electronic version available at: `http://www.mat.uni.torun.pl/preprints`, 18-2003.
12. Schreiber, T. (2004) Random dynamics and thermodynamic limits for polygonal Markov fields in the plane. Electronic version available at: `http://www.mat.uni.torun.pl/preprints`, 17-2004.

Fingerprint Image Segmentation Method Based on MCMC&GA*

Xiaosi Zhan[1], Zhaocai Sun[2], Yilong Yin[2], and Yun Chen[1]

[1] Computer Department, Fuyan Normal College, 236032, Fuyang, China
xiaoszhan@263.net, chenyun.fync@163.com
[2] School of Computer Science & Technology, Shandong University, 250100, Jinan, China
sunnykiller@126.com, ylyin@sdu.edu.cn

Abstract. Fingerprint image segmentation is one key step in Automatic Fingerprint Identification System (AFIS), and how to do it faster, more accurately and more effectively is important for AFIS. This paper introduces the Markov Chain Monte Carlo (MCMC) method and the Genetic Algorithm (GA) into fingerprint image segmentation and brings forward a fingerprint image segmentation method based on Markov Chain Monte Carlo and Genetic Algorithm (MCMC&GA). Firstly, it generates a random sequence of closed curves, which is regarded as the boundary between the fingerprint image region and the background image region, as Markov Chain, which uses boundary curve probability density function (BCPDF) as the index of convergence. Then, it is simulated by Monte Carlo method with BCPDF as a parameter, which is converged at the maximum. Lastly, Genetic Algorithm is introduced to accelerate the convergent speed. In conclusion, the closed curve with the maximum value of the BCPDF is the ideal boundary curve. The experimental results indicate that the method is robust to the low-quality finger images.

1 Introduction

In recent years, the technology of Automatic Fingerprint Identification has caused people's extensive concerns [1~5]. Fingerprint image segmentation is a key problem in fingerprint image processing and it is also one of the most intensively studied areas in fingerprint identification system. It is important for AFIS that the fingerprint image is segmented faster, more accurately and effectively.

The present fingerprint image segmentation methods can be summed up two specials: one is based on block-level [2,3], the other is based on pixel-level [4,5]. Both designed the algorithms according to the statistical character (e.g. Variance, Mean) of the gray fingerprint image. Yin Y.L. et al used the model of quadratic curve surface to carry out the fingerprint image segmentation [6], which regarded the gray variance,

* Supported by the National Natural Science Foundation of China under Grant No. 06403010, Shandong Province Science Foundation of China under Grant No.Z2004G05 and Anhui Province Education Department Science Foundation of China under Grant No.2005KJ089.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 391 – 398, 2005.

the gray mean and the orientation coherence as the parameters of the model. Generally, it satisfies the demand of the fingerprint image segmentation processing in common cases, but the result is unsatisfied when the fingerprint images are stronger peeled or have stronger interference by the prior remainder image.

Over the past 40 years, Markov Chain Monte Carlo (MCMC) method has penetrated many subjects, such as statistical physics, seismology, chemistry, biometrics and protein folding, as a general engine for inference and optimization [7]. In computer vision, Zhu S C et al. has done many works but not fingerprint image [8,9,10]. He Y L et al regarded the fingerprint image as Markov Random Field and carried out the fingerprint image segmentation successfully [11]. But, it can only generate the boundary curve only where the edge contrast between fingerprint and background is stronger, and it is unsatisfied when the interference of background is stronger.

The paper takes the closed curve as research object and proposes the fingerprint image segmentation method based on MCMC&GA. Firstly, it randomly generates Markov chain of closed curves. Then, it is simulated by Monte Carlo method to converge at the boundary curve that has the biggest value of the boundary curve probability density function. Lastly, Genetic Algorithm is introduced to accelerate the convergent speed.

2 Fingerprint Boundary Curve Markov Chain

2.1 The Model of Boundary Curve Probability

To a fingerprint image, if the closed curve, which can separate the fingerprint image region from background image region, can be found out, we think that the fingerprint image will be segmented well. We call this closed curve as boundary curve. As Fig.1 showed, the curve B successfully separates the fingerprint image region from the background image region and reduce the disturbance cause of the remainder and peeling image region at the same time, which the curves A, C, D can't accomplish. Hence, the curve B can be regarded as the boundary curve of this fingerprint image while the curve A, C and D can't be regarded as the boundary curve. So, the process of fingerprint image segmentation is to look for the boundary curve. If we can calculate the probability that a closed curve is the boundary curve of the fingerprint image according to the gray level of the fingerprint image, the closed curve with the biggest probability can be regarded as the boundary curve, for example, curve B as fig.1 showed. Obviously, such boundary curve probability density function (BCPDF) is required to satisfy the following conditions:

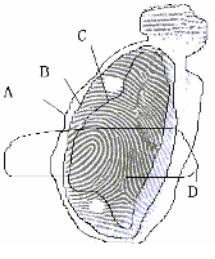


Fig. 1. Fingerprint image and the boundary curve

(1) The value of BCPDF of the closed curve in background image region (e.g. curve A) is less than that of the boundary curve (e.g. Curve B).

(2) The value of BCPDF of the closed curve within fingerprint image region (e.g. curve C) is less than that of the boundary curve (e.g. Curve B).

(3) The value of BCPDF of the closed curve that has crossed fingerprint image region and background image region (e.g. curve D) is less than that of the boundary curve (e.g. Curve B).

If we define inward ring and outward ring of a closed curve as fig.2 showed, compared with curves A, C, D, the boundary curve like curve B is required:

(1) The outward ring of the boundary curve is in background image region exactly. (\\\ denoted as outward ring as fig.2 showed).

(2) The inward ring of the boundary curve is within fingerprint image region exactly. (/// denoted outward ring as fig.2 showed).

In this paper, we denote the outward background probability density function of a closed curve Γ as $Pout(\Gamma)$ and the inward fingerprint probability density function as $Pin(\Gamma)$. Then, the value of $Pout(\Gamma)$ is the probability that the outward ring is in the background region and the value of $Pin(\Gamma)$ is the probability that the inward ring is in the fingerprint image region. So, if we denote BCPDF of Γ as $PL(\Gamma)$, then, we have:

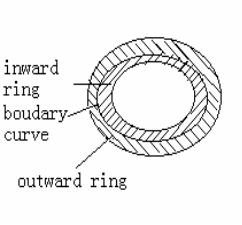


Fig. 2. Boundary curve and the inward ring

$$PL(\Gamma) = Pin(\Gamma)Pout(\Gamma)$$

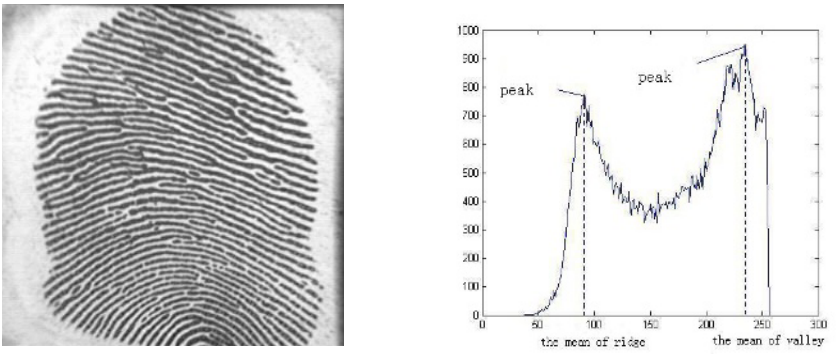
The following issue of fingerprint image segmentation is to find the closed curve Γ whose BCPDF value $PL(\Gamma)$ is the biggest. And, the latter work is to calculate the outward background probability $Pout(\Gamma)$ and the inward fingerprint probability $Pin(\Gamma)$.

2.2 Calculation for the Outward Background Probability Density and the Inward Fingerprint Probability Density

Generally, fingerprint image can be segmented into two kinds of regions as background image region and fingerprint image region, which also can be labeled as ridge region and valley region. If we study the fingerprint image carefully, we can find that the gray levels of pixels in ridge region are very close, and so as valley region and background region. Moreover, gray levels of pixels in valley region or in background region are so close. Hence, the result is that the gray level will gather into two domains and there are two peaks in the corresponding histogram, as fig.3 showed. The gray level where pixels in background region or in valley region gathered is called as the mean of valley, so as the mean of ridge. Then, it can be considered that pixels in ridge region obey the normal distribution with the mean of ridge as the form:

$$p(i(x,y) \mid ridge) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(g_m-\mu_l)^2}{2\sigma^2}} \quad (1)$$

Where g_m denotes the gray level of the pixel i(x,y), μ_l denotes the mean of ridge, σ^2 denotes variance. And, pixels in background region or in valley region obey the normal distribution with the mean of valley as the form:



(a) (b)

Fig. 3. The fingerprint image and the histogram, (a) is the original fingerprint image and (b) is the corresponding histogram

$$p(i(x,y) \mid valley) = p(i(x,y) \mid back) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(g_m-\mu_h)^2}{2\sigma^2}} \quad (2)$$

Where μ_h denotes the mean of valley.

To a closed curve Γ, if the outward ring of Γ is in the background image region then every pixel is in the background image region and obeys the normal distribution with the mean of valley. In conclusion, the probability of the outward ring of Γ is in the background image region completely can be written as the form:

$$Pout(\Gamma) = \prod_{m=1}^{k} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(g_m-\mu_h)^2}{2\sigma^2}} \quad (3)$$

Where k denotes the sum of pixels in the outward ring, g_m denotes gray level of the pixel i(x,y), μ_h denotes the mean of valley.

But to the fingerprint image region, pixels in it are either in ridge region or in valley region. The gray distribution of valley region is the same to that of background region. Hence, according to whether the pixels are in ridge region or not, we judge the inward ring of Γ being in the fingerprint image region or not. We can see, the ridge line and the valley line are always appear by turns in the fingerprint image region. Hence, it can be considered that the sum of the pixels in ridge region is equal to that of pixels in valley region approximately. In other words, the sum of pixels in ridge region is equal to half of the sum of all pixels in fingerprint image region approximately. So, the inward fingerprint probability can be written as the form:

$$Pin(\Gamma) = (1-|1-\frac{2k}{N}|) \prod_{m=1}^{k} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(g_m-\mu_l)^2}{2\sigma^2}} \quad (4)$$

Where N denotes the sum of all pixels in inward ring of Γ, k denotes the sum of pixels in ridge region, g_m denotes the gray level of the pixel in ridge region

and μ_l denotes the mean of ridge. The left coefficient has guaranteed that the value of $PL(\Gamma)$ is the biggest only if the sum of pixels in ridge region is half of the sum of all pixels in fingerprint image region, which is also the peculiarity of fingerprint image.

Now, we can calculate BCPDF $PL(\Gamma)$ of any closed curve Γ in fingerprint image, through calculating the outward background probability $Pout(\Gamma)$ and the inward fingerprint probability $Pin(\Gamma)$.

$$PL(\Gamma) = Pin(\Gamma)Pout(\Gamma) \tag{5}$$

In fingerprint image, the closed curve with the biggest value of BCPDF is the optimum solution of the fingerprint image segmentation. So, it is required to find the closed curve with the biggest value of BCPDF. A simple thought is looking for all closed curves in fingerprint image and finding one with the biggest BCPDF value. But, it is impossible to look for all the closed curves. Hence, there must be some approximate methods to do it. Markov Chain Monte Carlo (MCMC) will solve this kind of problem well. Generally, it required two steps with MCMC: (1) Generating Markov Chain according to the needs of problem. (2) Solving it with Monte Carlo and looking for the approximate answer.

2.3 Markov Chain of Boundary Curve in Fingerprint Image

Supposing the sequence $\{\Gamma_1, \Gamma_2, \cdots \Gamma_n\}$ of the boundary curve in fingerprint image is a random Markov Chain, it can be known by the property of Markov Chain that $P(\Gamma_{i+1} \mid \Gamma_i) = P(\Gamma_{i+1} \mid \Gamma_1, \Gamma_2, \cdots, \Gamma_i)$. In other words, the next state of the boundary curve Γ_{i+1} depends only on the current state Γ_i, but not the historical state $\{\Gamma_1, \Gamma_2, \cdots, \Gamma_{i-1}\}$. There are two basic requirements for designing Markov chain dynamics. Firstly, it should be ergodic and aperiodic. Given any two closed curve Γ, Γ', the Markov chain can travel from Γ to Γ' in finite steps. Secondly, it should observe the stationary equation.

This requirement is often replaced in a stronger condition: the so-called detailed balance equations. Brownian motion is a common stochastic process. So we design Markov chain in Brownian motion method as following:

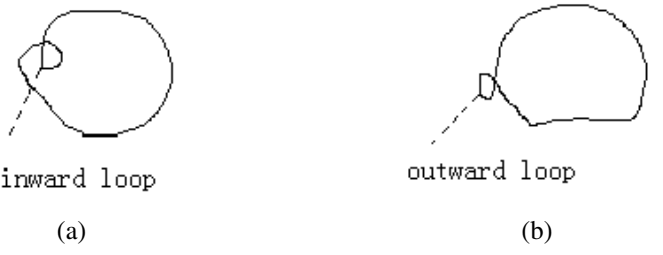


Fig. 4. Two kinds of lonops that need to be gotten rid of: (a) is the inward loop and (b) is the outward loop

Step1: Supposing Γ_i is the set of the point x_i^k, every point x_i^k does Brownian motion, $x_{i+1}^k = B(x_i^k)$.

Step2: Connect points x_{i+1}^k in turn. $\Gamma_{i+1}^0 = \{x_{i+1}^1, x_{i+1}^2, \ldots, x_{i+1}^k, \cdots, x_{i+1}^{n_i}\}$

Step3: Make up the collected curve and get rid of repeated loops, as the fig.4 showed, $\Gamma_{i+1} = m(\Gamma_{i+1}^0)$.

3 Markov Chain Monte Carlo and Genetic Algorithm

3.1 Monte Carlo Simulation

To the curve Markov chain, as 2.3 discussed, we should introduce some optimize solutions to obtain the value of BCPDF in view of the calculation speed. Monte Carlo is such method of simulation that can converge at the boundary curve quickly. The key of Monte Carlo method is the selection of the kernel. Here we apply the Metropolis-Hastings scheme that has the following function:

$$P(\Gamma_{i+1} \mid \Gamma_i) = \min\left\{1, \frac{PL(\Gamma_{i+1})}{PL(\Gamma_i)}\right\}. \tag{6}$$

To any state Γ_i and the next state Γ_{i+1} of Markov chain, we calculate the shell and decide if transform or not based on the value of the transform probability $P(\Gamma_{i+1} \mid \Gamma_i)$. Now, we can summarize that we do it by MCMC method in the following steps.

Step1: Generate a potential Markov chain as 2.3 illustrated.

Step2: Supposing the current state is Γ_i and the next state is Γ_{i+1} in Markov chain as step1, calculate the shell $P(\Gamma_{i+1} \mid \Gamma_i)$ as formula (6).

Step3: Generate a random variance u with uniform distribution at domain [0,1].

Step4: If $P(\Gamma_{i+1} \mid \Gamma_i) \geq u$, Markov chain go to the next state Γ_{i+1} and go to step 2.

Step5: If $P(\Gamma_{i+1} \mid \Gamma_i) < u$, Markov chain refuse to transform, $\Gamma_{i+1} = \Gamma_i$, and go to step2.

Step6: If the number of continue repeated transforms is more than 50, stop the process and consider the current answer is the optimum answer.

3.2 Genetic Algorithm

Because the Boundary Curve Markov Chain is generated by Brownian Motion, the next state may be very irregular. The price of convergent speed is big. The Genetic Algorithm will solve it. A genetic algorithm is a heuristically guided random search technique that concurrently evaluates thousands of postulated solutions. Biased random selection and mixing of the evaluated searches is then carried out in order to progress towards better solutions.

According to the Genetic Algorithm, the current closed boundary curve Γ_i can be divided into N divisions $D_i^k, k = 1,2,\cdots N$ and code it with number "0", correspondingly, the next N divisions D_{i+1}^k is coded with "1". If we randomly select code "0" or "1" to combine the codes (d,d,…,d), d is "0"or "1", the probability will be 2N.In other words, the probability of the new connected curve Γ_{i+1}^j is 2N. Then, from the 2N curves, we randomly select M curves $\Gamma_{i+1}^{j_k}, k = 1,2,\ldots,M$ and calculate the BCPDF $PL(\Gamma_{i+1}^{j_k})$, lastly, we select the curve $\Gamma_{i+1}^{j_k}$ which has the biggest BCPDF value as the next Markov state Γ_{i+1}. The steps of MCMC&GA is,

Step1: Generate a potential Markov Chain as 2.3 illustrated.

Step2: optimize the Markov Chain with Genetic Algorithm, as3.2 showed.

Step3: Monte Carlo simulates the optimized Markov Chain to the convergent solution Γ , as 3.1showed.

4 Experiment Results and Conclusion

To examine the effect of the algorithm proposed in the paper, we chose DB_B in the public data–BVC2004, which is considered the most hard fingerprint image database to segmented in common and disposed it with MCMC and MCMC&GA respectively.

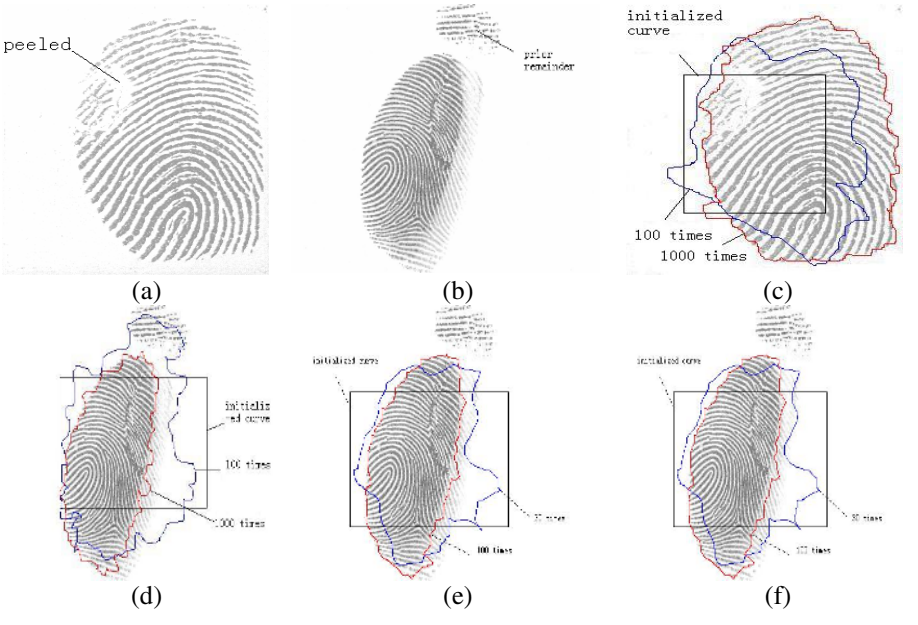


Fig. 5. Some fingerprint images and the segmentation results by MCMC and MCMC&GA ((a) and (b) are the original low-quality fingerprint images; (c) and (d) are the corresponding segmentation results of MCMC to (a) and (b) respectively; (e) and (f) are the corresponding segmentation results of MCMC&GA to (a) and (b) respectively).

Fig.5 is the representative fingerprint images and the fingerprint image segmented results. The (a) and (b) are the original fingerprint images with strong noise, the (c) and (d) are the segmentation results with MCMC when allowed to run for 100 iterations and 1000 iterations, the (e) and (f) are the results with MCMC&GA for 30 iterations and 100 iterations. The experimental results indicate that MCMC&GA is more effective.

The paper proposes the method of fingerprint image segmentation based on MCMC&GA. It takes the closed curve in fingerprint image as research object, randomly generates Markov chain of closed curves, then, it is simulated by Monte Carlo method to convergent to the boundary curve whose boundary curve probability function is the biggest.

The primary experimental results indicate that MCMC&GA method is robust to the fingerprint image with stronger disturbance of background, especially to the peeled fingerprint and the prior remainder fingerprint.

References

1. Jain, A. K., Uludag, U., Hsu R.L.: Hiding a Face in a Fingerprint Image. Proc. ICPR, Quebec City (2002): 756-759
2. Zhan, X.S.: Research on Several Key Issues Related to AFIS Based on Verification Mode. Ph.D Dissertation, Najing University (2003)
3. Jain, A.K., Hong, L., Bolle, R.: On-Line Fingerprint Verification. IEEE Transactions on Pattern Analysis and Machine Intelligence (1997): 302-314
4. Mehtre, B.M., Murthy, N.N., Kapoor, S., Chatterjee, B.: Segmentation of Fingerprint Images Using the Directional Images. Pattern Recognition (1987):429-435
5. Mehtre, B.M., Chatterjee, B.: Segmentation of fingerprint images-a composite method. Pattern Recognition (1995): 1657-1672
6. Yin,Y.L, Yang, X.K., Chen, X., Wang, H.Y.: Method based on Quadric Surface Model for Fingerprint Image Segmentation, Defense and Security, Proceedings of SPIE (2004):417-324
7. Green P.J.: Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination, Biometrika (1995):711-732
8. Zhu, S.C., Zhang, R., Tu, Z.W.: Integrating Bottom- Up/ Top- Down for Object Recognition by Data Driven Markov Chain Monte Carlo . Proc. IEEE Conference on Computer Vision and Pattern Recognition USA : Hilton Head Island (2000):738 -745
9. Tu, Z.W., Zhu, S.C., Shum, H.Y.: Image Segmentation by Data Driven Markov Chain Monte Carlo. Proc, ICCV 2001. Eighth IEEE International Conference on Computer Vision. Canada : Vancouver Vol.2 (2001):131 - 138
10. Tu, Z.W., Zhu, S.C.: Parsing Images into Region and Curve Processes[EB/OL]. http:/ /www.stat.ucla.edu/~ztu/ DDMCMC/curves/ region - curve.htm (2002)
11. He, Y.L., Tian, J., Zhang, X.P.: Fingerprint Segmentation Method Based on Markov Random Field. Proceedings of the 4th China Graph Conference (2002): 149-156

Unsupervised Segmentation of Text Fragments in Real Scenes

Leonardo M. B. Claudino[1], Antônio de P. Braga[1],
Arnaldo de A. Araújo[2], and André F. Oliveira[2]

[1] Centro de Pesquisa e Desenvolvimento em Engenharia Elétrica,
Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brazil
{claudino, apbraga}@cpdee.ufmg.br
[2] Depto. de Ciência da Computação, Universidade Federal de Minas Gerais,
Belo Horizonte, Minas Gerais, Brazil
{arnaldo, fillipe}@dcc.ufmg.br

Abstract. This paper proposes a method that aims to reduce a real scene to a set of regions that contain text fragments and keep small number of false positives. Text is modeled and characterized as a texture pattern, by employing the QMF wavelet decomposition as a texture feature extractor. Processing includes segmentation and spatial selection of regions and then content-based selection of fragments. Unlike many previous works, text fragments in different scales and resolutions laid against complex backgrounds are segmented without supervision. Tested in four image databases, the method is able to reduce visual noise to 4.69% and reaches 96.5% of coherency between the localized fragments and those generated by manual segmentation.

1 Introduction

Text fragments are blocks of characters (e.g. words and phrases) that often appear isolated from one another in scenes containing objects such as traffic signs, billboards, subtitles, logos, or car license plates. Such fragments are visually salient, especially due to high-contrast against the background, spatial properties, and occurrence of vertical borders.

This paper is particularly motivated by the problem of finding vehicular license plates in a scene, for plate recognition. The authors of [1], for instance, introduce a technique for finding license plates based in the supposition that the lines containing the plate have regular gray scale intervals and produce a signature of the plate. In [2], it is noted that there is a significant amount of vertical edges in the region of the license plate. The image is split in equally spaced horizontal lines and, for each line, the vertical edges are tagged when the difference of values is above a given threshold. The regions are formed by merging vertically adjacent tags and each region is a candidate plate.

The work of Mariano et al. [3] looks for evidences of text in a vehicle and is intended to support surveillance and security applications. The method produces clusters in L*a*b* space and each group is tested to decide whether it has pixels

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 399–406, 2005.

that belong to text. Neighbor lines are streaked to indicate the occurrence of pixels in candidate text clusters.

Clark et al. [4] look especially for paragraphs and large blocks of text, and propose five local statistical metrics that respond to different text attributes. The metrics are combined in a three-layer neural network trained with patterns from 200 text and 200 non-text regions extracted from 11 manually labeled images. The paper presents only a few qualitative results and shows that text is found in different scales and orientations.

In the work of Wu et al. [5], the image suffers texture segmentation by applying three second order gaussian derivatives in different scales followed by a non-linear function. Then, k-means (with $k = 3$) is applyed. Post-processing consists in forming and clustering regions in three different scales, false detection reduction and text refinement. Finally, the text is binarized so that it can be fed to a character recognizer. The work assumes that text appears horizontal in the image.

In the following section, it is presented a new unsupervised text segmentation technique. Here, to be considered a text fragment, a region must satisfy three main conditions:

- being at most constituted from vertical edges or borders;
- being long, with two or more characters;
- presenting regularity of vertical borders throughout it's extention.

Like [5,6,7], it models and characterizes text as a texture pattern. In the novel approach, segmentation is done keeping only the vertical detail coefficients images, which are blurred and then binarized. After that, eight-connected binary regions are selected according to its spatial behavior. The resulting regions are mapped back to spatial domain, becoming candidate text fragments. The expected text fragments are selected according to two proposed content-based features, in a final step.

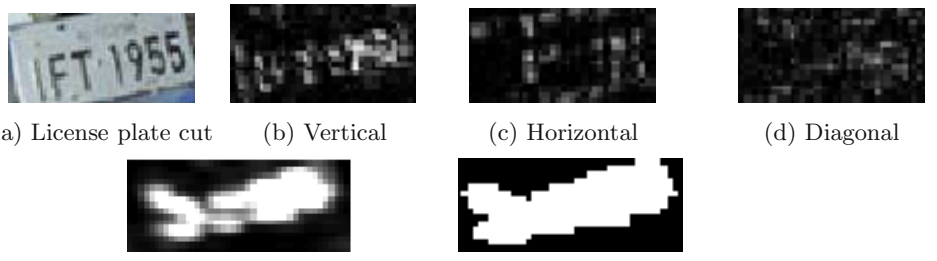
2 Proposed Method

The proposed solution is divided in three procedures. After applying the wavelet transform to the input image and extracting the vertical detail in three different scales, regions are segmented and then selected into candidate fragments, based on spatial aspects. Finally, only those fragments satisfying certain content requirements are considered valid text fragments and output by the method.

2.1 Wavelet Transform

This work adopted the discrete wavelet transform (DWT) and thus brought the process of fragment segmentation to the spatial-frequency (s-f) domain.

A Quadrature Mirror Filter (QMF) bank [8] with three levels of resolution is applied to the input image, with Daubechies-4 being used as the basis function.



(a) License plate cut (b) Vertical (c) Horizontal (d) Diagonal

(e) Cut (b) after blurring. (f) Cut (e) after thresholding.

Fig. 1. Cuts at the text area (license plate) of one of the sample vehicle images: original image (a), vertical (b), horizontal (c), diagonal (d) normalized sub-bands, at the higher decomposition level corresponding image. Vertical coefficients after blurring (e) and thresholding (f).

Only the images of vertical (sub-band) detail coefficients are kept, as they give more information and less noise, compared to the horizontal and diagonal ones (Fig. 1). The vertical coefficients image output from the $p-th$ decomposition level, $\mathbf{V}(p)$, of size $m \times n$ is normalized w.r.t. the sub-band energy $E(\mathbf{V}(p))$, yielding $\mathbf{V}^N(p)$.

$$\mathbf{V}_{ij}{}^N(p) = \frac{[\mathbf{V}_{ij}(p)]^2}{E(\mathbf{V}(p))}, \quad where \tag{1}$$

$$E(\mathbf{V}^N(p)) = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} [\mathbf{V}_{ij}(p)]^2 \tag{2}$$

2.2 Region Segmentation

Region segmentation is performed as an unsupervised clustering on the coefficient images generated in the previous step. This is done by first convolving two unidimensional gaussian masks (a vertical and a horizontal one) and the image, resulting in the filtered image $\mathbf{F}$. The size of the two gaussian windows are kept the same for the three images, which favors the production of larger regions, as decomposition level increases.

The role of the filtering is to group neighboring detail coefficients. This procedure is important because it merges, for instance, fragment coefficients from defective or incomplete characters, or also coefficients from characters that belong to different lines. A binarization threshold computation follows, taking into account the global response of the image to the vertical details. Thus, the threshold θ tells whether a point in $\mathbf{F}$ is salient or not, and is defined in (3).

$$\theta = \mu(\mathbf{F}) + \frac{\sigma(\mathbf{F})}{2} \tag{3}$$

The value $\mu(\mathbf{F})$ stands for the mean of filtered image $\mathbf{F}$, and $\sigma(\mathbf{F})$ for it's standard deviation. Fig. 1 also shows that after gaussian blurring and further binarization the vertical detail coefficients corresponding to the license plate text were properly grouped. The regions are tracked according to the binary connectivity of the component pixels to its eight neighbors. Each of them has its position and bounding-box calculated.

2.3 Spatial Selection of Segmented Regions

First of all, the orientation (or rotation angle) of each segmented region is estimated by performing a linear regression (least-squares) on its binary image pixels, each contributing a pair (x, y).

The bounding-box is rotated about its center, producing the corrected one. The measures w and h are the greatest intervals of the region along x and y, thus corresponding to its real width and height, respectively. The aspect ratio of the region is calculated from the ratio between w and h.

In the next step, spatial coordinates x (first column), y (first line), w and h are mapped back to spatial domain (x^e, y^e, w^e and h^e) following Eqs. 4 and 5, where p corresponds to the current decomposition level.

$$x^e(x) = 2^p \cdot x - \sum_{i=1}^{p} 2^i, \quad y^e(y) = 2^p \cdot y - 3 \cdot \sum_{i=1}^{p} 2^i \tag{4}$$

$$w^e(x) = 2^p \cdot w, \quad h^e(y) = 2^p \cdot h \tag{5}$$

The procedure represented by (4), for each already performed decomposition, doubles the coordinates and subtracts $0.25 \cdot f = 2$ for x and $0.75 \cdot f = 6$ for y, being f the lenght of the decomposition filters adopted ($f = 8$). Values w and h, however, are just doubled.

After spatial characterization, each region is evaluated and must have at least 10 pixels of h^e, values of w^e greater than h^e, and an aspect ratio > 2.

2.4 Content-Based Selection of Candidate Text Fragments

Fragments of text usually have high density of vertical edges, regularly placed from line to line [1,2]. Actually, the second observation is true only if the text is aligned with the capturing device. So, being avaliable well-bounded, horizontally oriented (or rotation-fixed) regions, it is reasonable to suppose that the lines containing text are those with greatest edge density and take a considerable area of the candidate region. The edges in those lines should be also distributed regularly, that is, their central position should not change much from line to line.

In order to describe a candidate fragment according to those text content hypothesis, a simple technique is proposed. It starts by determining the occurrence of relevant transitions in the rotated candidate fragment image. A transition is relevant if the absolute difference of intensity between two co-linear pixels is greater than a percent threshold, k_{MIN}, relative to the greatest difference observed in the whole image. For each line, the transitions are inspected and stored. If the transitions in a line are mostly concentrated before or after the middle of the fragment, that line is discarded, since text characters must occur along the whole line and so must do the corresponding transitions. The algorithm then groups valid, consecutive lines into blocks. The blocks are separated by valleys in the transition profile (Fig. 2) and the block containing more lines is considered to be where the text fragment more probably is. After the most probable block

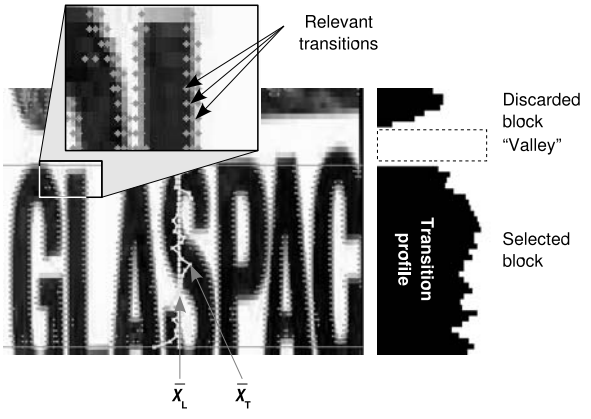


Fig. 2. Relevant transitions and each line transition profile, in a candidate fragment present in one of the test samples

is pointed out, a measurement of the regularity of its transitions, the ρ feature, is calculated, according to (6).

$$\rho = min(\frac{\sum_i |\bar{x}_T(i) - \bar{x}_L|}{\sum_i n_T(i)}, 1), where \tag{6}$$

$$\bar{x}_L = \frac{\sum_{i=1}^{n_L} \bar{x}_T(i)}{n_L} \tag{7}$$

$$\bar{x}_T(i) = \sum_{k=1}^{n_T(i)} \frac{t_i(k)}{n_T(i)} \tag{8}$$

In (8), $x_T(i)$ is calculated as the average x-coordinate of the transitions in line i. In (7), $\bar{x}_L$, the central x-coordinate, is the average of $\bar{x}_T(i)$, for each of the n_L lines as shown in Fig. 2. The value ρ is the integral of the differences of the average position of each line and the central line. It is divided by the total number of transitions in the block, n_T, that appears as the area of the profile also shown in Fig. 2, to quantify the importance of the difference. Since $n_T(i)$ is not an exact normalizer, ρ is saturated at 1.

Another extracted feature is the ratio between the total transitions in the block and total transitions in the fragment it belongs to, n_F, calculated according to (9).

$$a = \frac{\sum_{i=1} n_T(i)}{n_F} \tag{9}$$

The final step in content-based selection is to decide whether the pairs of features (ρ, a) extracted from each of the candidate fragments are to be considered as belongig to a text fragment. For simplicity, here they are only compared to pre-defined thresholds ρ_{MAX} and a_{MIN}. The final results of the method are illustrated in Fig. 3.

(a) Oriented selected regions. (b) Image reduction.

Fig. 3. Content-based selection of candidate fragments into text fragments. Parameters used: $\rho_{MIN} = 0.2$ and $a_{MIN} = 0.95$).

3 Results and Conclusions

A total of 580 images from four databases were tested (Tab. 1). The results produced by the method were compared to manual segmentation made by three collaborators that marked the bounding boxes of text fragments (or text blocks) in the scenes. The parameters employed were $\rho_{MIN} = 0.35$ and $a_{MIN} = 0.85$. Results were evaluated according to two indicators, calculated after the execution of each of the three phases of the method. The first, true positives (I_{TP}), quantifies the accuracy of the method in terms of fragment finding: the returned fragments are intersected with the manually selected region and the resulting area is divided by the total marked area of the manually selected region. The second indicator, false positives (I_{FP}), evaluates the capacity of removing distractors from the image: the area of returned fragments that do not correspond to the manually selected region is divided by the total area of the image.

Table 1. Summary of image databases employed in the experiment

A. Images of 363 vehicles with visible license plates. Acquired using two digital cameras under different environmental conditions. Plate text in different scales and orientations. Sampling resolution of 320 × 240 pixels. Compressed JPEGs.
B. 100 images of vehicles with visible license plates. Acquired from various websites. Diverse image sizes and file formats, mostly compressed JPEGs.
C. 88 images of vehicles with visible license plates. Acquired from campus surveillance MPEG-compressed videos, the camera is sometimes out of focus. Sample resolution of 640 × 480 pixels.
D. 29 images with diverse text fragments. Includes advertisements, banners, historical documents and office room number plates. Various sizes and file formats.

For database A, 98.21% of the regions of interest are detected by the first phase of the method and 91.82% remain after the region selection stages. Meanwhile, the false positives drop from initial 26.42% to about 5.37%, an average reduction of 20% of the image area. Figs. 4 (a) and (b) present the results for an example from database A and it can be seen that the method deals well with varying perspective. Figs. 4 (c) and (d) show the results for a scene from image

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4. Qualitative results after testing the method in the four image databases

database B. The amount of true positives for this database falls from 96.65% to 91.04%, while false positives decrease from 34.83% to 5.79%. In Figs. 4 (d) and (e), the method finds both the vehicle's license plate and the taxi mark. Keeping 96.50% from the 98.06% of true positives is a very good output for database C,

since it is negatively affected by high video compression and bad focusing of the camera. The decrease in false positives is also high, reaching 4.69%. Figs. 4 (f) and (g) depict the results for one from the 29 available images of database D, a low resolution flatbed scanned historical document image. In that database, text appears with greatest variability. The indicators show a regular true positive rate (73.92% after all) and spurious regions removal around 7.78%.

The method presented here succeeds the proposed goals, since it is demonstrated by I_{TP} and I_{FP} that it reaches 96.50% of true positives and reduces the visual noise to 4.69%, according to manual segmentation. Now that text fragments in arbitrary scenes are efficiently detected by the presented method, it will be integrated to a character recognition system that will operate on the fragments it outputs.

References

1. J. Barroso, A. Rafael, E.L.D., Bulas-Cruz, J.: Number plate reading using computer vision. In: IEEE-International Symposium on Industrial Electronics ISIE'97. (1997) 761–766
2. Setchell, C.J.: Applications of Computer Vision to Road-traffic Monitoring. PhD thesis, Faculty of Engineering, Department of Electrical and Electronic Engineering of the Univesity of Bristol (1997)
3. Mariano, V.Y., Kasturi, R.: Detection of text marks on moving vehicles. In: 7th International Conference on Document Analysis and Recognition (ICDAR 2003), 2-Volume Set, 3-6 August 2003, Edinburgh, Scotland, UK. (2003) 393–397
4. Clark, P., Mirmehdi, M.: Finding text regions using localised measures. In Mirmehdi, M., Thomas, B., eds.: Proceedings of the 11th British Machine Vision Conference, BMVA Press (2000) 675–684
5. Wu, V., Manmatha, R., Riseman, E.M.: Textfinder: An automatic system to detect and recognize text in images. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 1224–1229
6. K. Etemad, D.D., Chellapa, R.: Multiscale segmentation of unstructured document pages using soft decision integration. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 92–96
7. Jain, A.K., Yu, B.: Automatic text location in images and video frames. Pattern Recognition **31** (1998) 2055–2076
8. Smith, J.R.: Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression. Phd. thesis, Graduate School of Arts and Sciences, Columbia University, New York, NY. (1997)

A New Efficient Method for Producing Global Affine Invariants

Esa Rahtu[1], Mikko Salo[2], and Janne Heikkilä[1]

[1] Machine Vision Group, Department of Electrical and Information Engineering, P.O. Box 4500, 90014 University of Oulu, Finland
{erahtu, jth}@ee.oulu.fi
[2] Rolf Nevanlinna Institute, Department of Mathematics and Statistics, P.O. Box 68, 00014 University of Helsinki, Finland
msa@rni.helsinki.fi

Abstract. This paper introduces a new efficient way for computing affine invariant features from gray-scale images. The method is based on a novel image transform which produces infinitely many different invariants, and is applicable directly to isolated image patches without further segmentation. Among methods in this class only the affine invariant moments have as low complexity as our method, but as known they also possess many considerable weaknesses, including sensitivity to noise and occlusions. According to performed experiments it turns out that our novel method is more robust against these nonaffine distortions observed in image acquisition process, and even in practice its computation time is equivalent to that of the affine invariant moments. It is also observed that already a small subset of these new features is enough for successful classification.

1 Introduction

Recognizing the contents of images has long been one of the main topics in computer vision research. This task is very challenging since already the results of the image acquisition process are affected by many factors making the actual recognition process even more complicated. One of the key problems rising during the imaging process are the geometric transformations caused by the distortions in the pose of the objects. A sophisticated solution is to find a set of descriptors invariant to these deformations, which then provide preferable inputs for classification and recognition algorithms.

Different types of invariants have been under research already for quite some time, but not so many of the introduced methods are capable of producing invariants in complex cases like affine and projective transformations. However, these are often just the models we need to give adequate approximations for real photographing situations. In addition most of the methods capable of handling these transformations are based on salient points or contours found using some other, possibly error prone, techniques.

Global affine invariant methods, capable of producing affine invariant features computed directly from gray-scale images, include affine invariant moments

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 407–414, 2005.

(AIM) [1,2], cross-weighted moments [3], Ben-Arie's frequency domain technique [4], trace transform [5], and Multi-Scale Autoconvolution (MSA) [6,7]. Many of these introduce advanced ideas and give high accuracy, but usually at the expense of high complexity and computational demands. This makes them impractical in systems with very limited amount of processing capacity, not to speak of real time systems. Only the affine invariant moments make an exception, and in fact in terms of computational speed AIMs are clearly superior to any other method in this class. However they do have considerable weaknesses, including sensitivity to nonaffine deformations like noise, occlusions, and nonlinearities, making the use of this method in many applications impossible.

In this paper we propose a new method for producing global affine invariants which has the same complexity as affine invariant moments. The proposed novel image transform immediately gives infinitely many different invariants, and according to experiments it turns out to be more robust with respect to noise and occlusions as affine invariant moments. Our method has some similarities with recently proposed Multi-Scale Autoconvolution, but is significantly faster to compute.

2 Motivation

We begin with a short discussion of the Multi-Scale Autoconvolution (MSA) transform [6], which will motivate a new affine invariant transform in Section 3. Let $f : \mathbf{R}^2 \to \mathbf{R}$ be an image function. The Fourier form of MSA is

$$Mf(\alpha, \beta) = \frac{1}{\|f\|_{L^1}^3} \int_{\mathbf{R}^2} \hat{f}(-\xi)\hat{f}(\alpha\xi)\hat{f}(\beta\xi)\hat{f}(\gamma\xi)\, d\xi \tag{1}$$

where $\hat{f}(\xi) = \int_{\mathbf{R}^2} e^{-2\pi i x\cdot\xi} f(x)\, dx$ and $\gamma = 1 - \alpha - \beta$.

This form produces affine invariants: $M(f \circ \mathscr{A}^{-1}) = Mf$ for any affine transformation $\mathscr{A}$. This is due to two facts. If f is replaced by the translation $g(x) = f(x - x_0)$, then $\hat{g}(\xi) = e^{-2\pi i x_0\cdot\xi}\hat{f}(\xi)$, and the choice $\alpha + \beta + \gamma = 1$ makes the exponentials cancel in (1). Also, if f is replaced by $g(x) = f(A^{-1}x)$ where A is a 2×2 nonsingular matrix, then $\hat{g}(\xi) = |\det A|\hat{f}(A^t\xi)$, i.e. $\hat{g}$ is obtained from $\hat{f}$ essentially by composing with a matrix. If one then makes the change of variables $\xi \mapsto A^{-t}\xi$ in (1) everything cancels and we see that (1) is also invariant under composition with nonsingular matrices.

If f represents an $N \times N$ image then the computational complexity of evaluating (1) is essentially the same as for computing the FFT of f, which is $O(N^2 \log N)$. Although this is fairly good compared to many global affine invariant methods, it is still much slower than the $O(N^2)$ complexity of affine invariant moments.

3 Definition

We base our new method on the formula (1), but instead of using the Fourier transform we just use the function f. This will give an expression which is

invariant under composition with matrices, but one loses translation invariance. To deal with this we will normalize f by computing the centroid

$$\mu(f) = \frac{1}{\int_{\mathbf{R}^2} f(x)\,dx}\Big(\int_{\mathbf{R}^2} x_1 f(x)\,dx, \int_{\mathbf{R}^2} x_2 f(x)\,dx\Big)^t$$

and using the normalized function $\tilde{f}(x) = f(x + \mu(f))$.

Definition 1. *Let $f \in L^\infty(\mathbf{R}^2)$ be compactly supported. For $\alpha, \beta \in \mathbf{R}$ define*

$$If(\alpha,\beta) = \frac{1}{\|f\|_{L^1}} \int_{\mathbf{R}^2} \tilde{f}(x)\tilde{f}(\alpha x)\tilde{f}(\beta x)\,dx. \tag{2}$$

Proposition 1. *$I(f \circ \mathscr{A}^{-1}) = If$ for any affine transformation $\mathscr{A}$.*

Proof. We first show that I is translation invariant. If $g(x) = f(x - x_0)$ then $\|g\|_{L^1} = \|f\|_{L^1}$ and $\mu(g) = \mu(f) + x_0$. Thus $\tilde{g}(x) = g(x + \mu(g)) = \tilde{f}(x)$, and consequently $Ig = If$. Also, if A is a nonsingular matrix let $g(x) = f(A^{-1}x)$. Since $\mu(g) = A\mu(f)$ one has $\tilde{g}(x) = f(A^{-1}(x + A\mu(f)) = \tilde{f}(A^{-1}x)$ and

$$Ig(\alpha,\beta) = \frac{1}{|\det A|\|f\|_{L^1}} \int_{\mathbf{R}^2} \tilde{f}(A^{-1}x)\tilde{f}(\alpha A^{-1}x)\tilde{f}(\beta A^{-1}x)\,dx.$$

The change of variables $x \mapsto Ax$ gives that $Ig = If$.

Remark 1. One obtains more general global affine invariants in the following way. Let $H : \mathbf{R}^k \to \mathbf{R}$ be a measurable function, and define

$$I'f(\alpha_1,\ldots,\alpha_k) = \frac{1}{\|f\|_{L^1}} \int_{\mathbf{R}^2} H(\tilde{f}(\alpha_1 x),\ldots,\tilde{f}(\alpha_k x))\,dx. \tag{3}$$

If H satisfies some conditions (e.g. H continuous and $H(\bar{0}) = 0$), then the expression is well defined for compactly supported $f \in L^\infty(\mathbf{R}^2)$.

The proof of Proposition 1 shows that I' is also affine invariant. By changing variables $x \mapsto \alpha_1^{-1}x$ we see that no information is lost if we normalize $\alpha_1 = 1$. In (2) we chose $k = 3$ and $H(x,y,z) = xyz$ and made the normalization $\alpha_1 = 1$. This choice seems natural and it gave good results in the experiments, however we expect that other choices for H may be useful in affine invariant classification.

The transform $f \mapsto If$ has the following symmetries, which are easily obtained from (2) by changes of variables.

Proposition 2. (a) $If(\alpha,\beta) = If(\beta,\alpha)$,
(b) $If(\alpha,\beta) = \alpha^{-2}If(1/\alpha,\beta/\alpha)$ *for* $\alpha \neq 0$,
(c) $If(\alpha,\beta) = \beta^{-2}If(1/\beta,\alpha/\beta)$ *for* $\beta \neq 0$.

The symmetries may be used to show that it is enough to compute $If(\alpha,\beta)$ in the triangle $T = \text{convex hull}\,\{(-1,-1),(-1,1),(1,1)\}$. The values $If(\alpha,\beta)$ outside T may be computed from the values in T using the symmetries.

In Figure 1 we have illustrated two sample images and logarithms of their transforms in range $\alpha, \beta \in [-1,1]$. Notice the symmetries of the transform.

4 Completeness Issues

According to our knowledge none of the global affine invariant methods have yet been proved to be complete, i.e. that knowing all the invariant values would uniquely determine the image up to affine transformation. Despite this fact, in practice many of the proposed methods perform fine with natural images. There is some nonuniqueness with our new method, and we will discuss this next.

If we consider binary images which are starlike and symmetric with respect to their centroids, one can see that the values $If(\alpha, \beta)$ are the same for all these sets (a set K is starlike if for any $x \in K$ the line segment between the centroid and x lies in K). Another case arises if f and g are two images with centroid at the origin such that $f(x) = g(-x)$ in C and $f(x) = g(x)$ outside of C, where C is a two-sided sector (i.e. $tx \in C$ whenever $x \in C$ and $t \in \mathbf{R}$). Then $If = Ig$ even though the images may not be related by affine transformation.

The reference method, affine invariant moments, has a similar weakness if only low order moments are considered. If g is any $2M$ times continuously differentiable compactly supported function, it is easy to see that the functions f and $f + \partial_{x_1}^M \partial_{x_2}^M g$ have the same moments up to order $M-1$. This gives infinitely many images which are not related by affine transformation but which are not distinguished by moments up to order $M-1$. Usually the moments used in practice are only up to order 3.

It could be argued that in practice images which cause mentioned nonuniqueness arise rarely, and in many applications one may be able to use $If(\alpha, \beta)$ for successful affine invariant classification. However, in applications where classification accuracy is more important than speed, more sophisticated methods like Multi-Scale Autoconvolution may give better results.

5 Implementation and Computational Complexity

In this section we consider some issues in the implementation of $If(\alpha, \beta)$ and show that it has $O(N^2)$ complexity. The Matlab program we used for computing $If(\alpha, \beta)$ in our experiments in Section 6 is also available at website: http://www.ee.oulu.fi/research/imag/msa/.

To apply the method to digital images one needs a discrete form of the invariant, and this is done by discretizing the integral (2) resulting in

$$If(\alpha, \beta) = \frac{1}{\sum_{x,y} f(x,y)} \sum_{x,y} \tilde{f}(x,y)\tilde{f}(\alpha x, \alpha y)\tilde{f}(\beta x, \beta y). \tag{4}$$

To evaluate this we need a method to get $\tilde{f}(x, y)$ from the original image function $f(x, y)$. Recall that $\tilde{f}(x, y)$ is a translated version of original image $f(x, y)$ so that the origin of the image coordinates (x, y) coincides with the image centroid. The coordinates of the centroid however rarely are integers, and if we just translate the pixel values to their new coordinates the resulting image does not have its values in the integer grid anymore. In (4) we need samples on three grids centered at the origin and having sample intervals $\{1, \alpha, \beta\}$, and thus we need

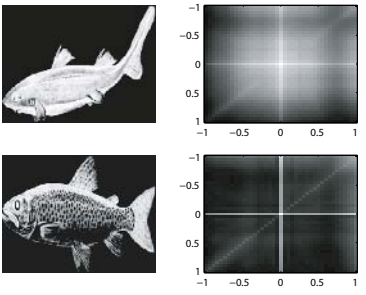


Fig. 1. Two images and logarithms of their transforms

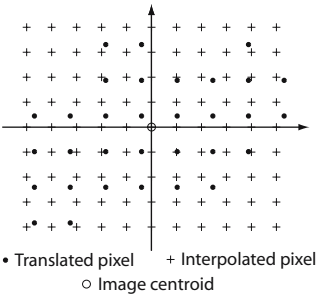


Fig. 2. Example of interpolation scheme

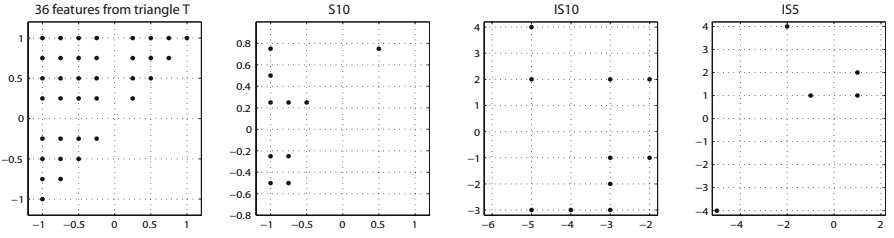


Fig. 3. (α, β) pairs used in the experiments

to interpolate them from the translated image. This is illustrated in Figure 2. It should be noted that one may select the interpolation grid so that it will have samples needed for all used (α, β) pairs, and we can achieve all desired invariants with only one interpolation. In our implementation we used bilinear interpolation as it seemed to perform well. After having the interpolated values the calculation of the resulting invariants is quite straightforward.

The symmetries in Proposition 2 indicate that we would only need to take (α, β)s from the triangle $\{(-1,-1),(-1,1),(1,1)\}$, and thus we chose to take uniform sampling with interval 0.25. However we also believe that already a smaller set would be enough for classification, and to test this we chose a subset of 10 pairs out of these 36. We will denote this invariant set by S10. In addition, if the chosen (α, β) pairs have integer values, the needed interpolation would be minimal reducing the computational demands. Hence we also took two sets of integer (α, β)s, one having 5 and other 10 pairs, addressed as IS10 and IS5 respectively. The three selected (α, β) sets are shown in Figure 3.

By looking at (4) one can directly see that the computational complexity is $O(N^2)$ for an $N \times N$ image. In addition finding $\tilde{f}(x, y)$ requires also interpolation to $N \times N$ points having $O(N^2)$ complexity, which results in an overall complexity $O(N^2)$. Furthermore it should be noted that with this same interpolation we are able to get also all other invariants that have parameters $(a \cdot \alpha, b \cdot \beta)$ where $a, b \in \mathbb{N}$. For affine invariant moments it quite straightforward to see that the resulting complexity is the same $O(N^2)$ for an $N \times N$ image.

6 Experiments

In this section we perform some experiments assessing our new method, implemented as described in Section 5. We compare the achieved results with affine invariant moments with 4, 10 and 60 independent invariants achieved using polynomials given in [2]. We will refer to these methods as AIM4, AIM10 and AIM60, respectively. For the experiment we took 94 gray scale images of fish, resolution 200×400, and used them to train a nearest neighbor classifier for each tested method. Then the classification performance was estimated using these same images disturbed by a random affine transformation combined with one of the following nonaffine distortions, Gaussian noise, occlusion, and nonuniform illumination. We also tested the performance in the presence of projective transformation. We now define how these distortions were created and illustrate some examples of these and the original images in Figure 4.

The Gaussian noise was uniformly distributed to whole image and then using thresholding removed from the background to eliminate the effect of the background size. The occlusion was generated by introducing different sized and randomly situated square shapes on the objects. The illumination distortion was done so that starting from the vertical center line, to the left we linearly decreased the gray-scale values and to the right we similarly increased them. The projective transformation is defined by the expression

$$\mathscr{P}(x) = \Big(\frac{p_{11}x_1 + p_{12}x_2 + p_{13}}{p_{31}x_1 + p_{32}x_2 + p_{33}}, \frac{p_{21}x_1 + p_{22}x_2 + p_{23}}{p_{31}x_1 + p_{32}x_2 + p_{33}}\Big),$$

where $p_{31}x_1 + p_{32}x_2 + p_{33} \neq 0$. In our experiment these parameters were chosen so that $p_{11} = p_{22} = p_{33} = 1$, $p_{12} = p_{13} = p_{21} = p_{23} = 0$, and the value $c = |p_{31}| + |p_{32}|$, which can be seen as a measure for the nonlinearity of the transformation, was varied.

To reduce the effect of randomness we performed each test, excluding projective transformation test, 1880 times and took the mean errors of these runs to be the resulting estimates. We have plotted these resulting error rates in Figure 5. From there it can be observed that our new method performs well under affine transformations and it seems to give better results than affine invariant moments

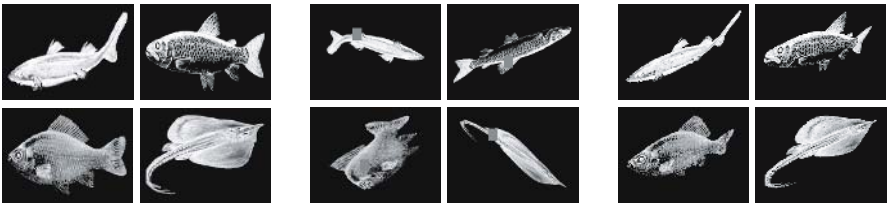


Fig. 4. Samples of fish images used in the experiments. On left there are 4 original images, in the middle 4 occluded images with occlusion size 30 and on the right 4 projective transformed images with nonlinearity value $2 \cdot 10^{-3}$.

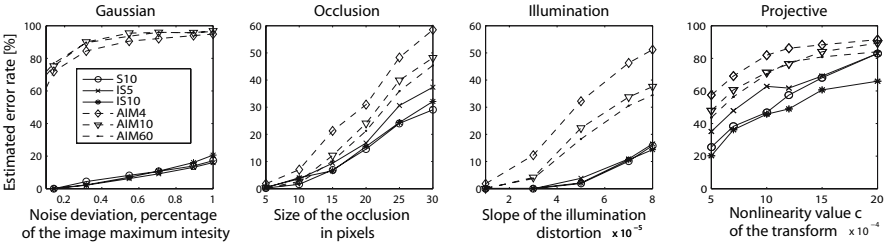


Fig. 5. The estimated error rates in different experiments with all tested methods

in all experiments. Especially in the Gaussian noise and nonuniform illumination tests the difference is very clear. A significant improvement can also be seen in the results of projective transformation test, thought this kind of distortion seems to be difficult to handle for the both of the tested methods. If we take a look at the different forms of our method, it can be noted that there are slight performance differences in some tests. In most cases the trend is that using more features we get slightly better results, especially in the projective transformation and occlusion tests. However this is not always the case and already 5 invariants give reasonable results. Thus one should select such features that best meet the overall requirements of the application. In addition we also believe that for small images one should not use very large (α, β), since then the scaled images would have only few nonzero pixels.

Comparing the affine invariant moments together, excluding the Gaussian noise test we get better results if we use more invariants, but the improvement seems to get quite small when we have already 10 invariants. In the Gaussian noise test the trend was to the opposite direction, but one can argue that the errors with any number of invariants are so high that the method is useless in such classification tasks anyway.

For final confirmation we performed an experiment to see how the actual computing times relate with these methods. We also added the cross-weighted moments with four invariants and MSA with 30 invariants to the same test to have some reference. We took an image having all pixels nonzero and measured the computation time for each method as a function of image size. In Figure 6 we have two plots, the first with results only of AIM, S10 and IS5, and the second

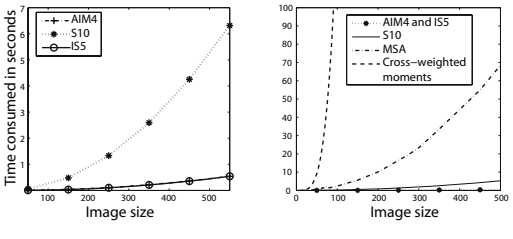


Fig. 6. The measured computation times for AIM4, S10, IS5, 4 cross-weighted moments and 30 MSA values

showing also two comparison methods. From these results it can be observed that our new method has also in practice the same computational complexity as affine invariant moments, depending a bit on which (α, β) pairs are used. However as shown by the classification experiments already the IS5 has better performance than AIMs. The results also illustrate how these two methods are clearly in their own category as far as computation time is concerned and this might be very crucial from the perspective of practical applications. We further note that these computations were made using Matlab and one can get even better performance using more optimized implementations with e.g. C language.

7 Conclusions

In this paper we have presented a novel efficient affine invariant image transform, which can provide a good alternative for affine invariant moments in applications where speed and low complexity are crucial. We assessed the new method in image classification tasks where it seemed to clearly outperform affine invariant moments, and it was observed that in many cases a small subset of all features is enough for successful classification. We expect that as the functionality of many mobile and small-scale systems increases, there is greater demand for fast and accurate image analysis methods. Thus we believe that our novel method can offer new possibilities for such applications.

Acknowledgments

The authors would like to thank the Academy of Finland (project no. 102083), and Prof. Petrou and Dr. Kadyrov for providing us the fish image database.

References

1. J. Flusser and T. Suk, "Pattern recognition by affine moment invariants," *Pattern Recognition*, vol. 26, no. 1, pp. 167–174, 1993.
2. T. Suk and J. Flusser, "Graph method for generating affine moment invariants," *Proc. International Conference on Pattern Recognition*, vol. 2, pp. 192–195, 2004.
3. Z. Yang and F. Cohen, "Cross-weighted moments and affine invariants for image registration and matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 804–814, 1999.
4. J. Ben-Arie and Z. Wang, "Pictorial recognition of objects employing affine invariance in the frequency domain," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 604–618, 1998.
5. M. Petrou and A. Kadyrov, "Affine invariant features from the trace transform," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 30–44, 2004.
6. J. Heikkilä, "Multi-scale autoconvolution for affine invariant pattern recognition," in *Proc. International Conference on Pattern Recognition*, pp. 119–122, 2002.
7. E. Rahtu and J. Heikkilä, "Object classification with multi-scale autoconvolution," in *Proc. International Conference on Pattern Recognition*, vol. 3, pp. 37-40, 2004.

Color Fourier Descriptor for Defect Image Retrieval

Iivari Kunttu[1], Leena Lepistö[1], Juhani Rauhamaa[2], and Ari Visa[1]

[1] Tampere University of Technology, Institute of Signal Processing
P.O. Box 553, FI-33101 Tampere, Finland
{Iivari.Kunttu, Leena.Lepisto,Ari.Visa}@tut.fi
http://www.tut.fi
[2] ABB Oy, Process Industry P.O. Box 94, FI-00381 Helsinki, Finland
Juhani.Rauhamaa@fi.abb.com
http://www.abb.com

Abstract. The shapes of the objects in the images are important in the content-based image retrieval systems. In the contour-based shape description, Fourier descriptors have been proved to be effective and efficient methods. However, in addition to contour shape, Fourier description can be used to characterize also the color of the object. In this paper, we introduce new Color Fourier descriptors. In these descriptors, the boundary information is combined with the color of the object. The results obtained from the retrieval experiments show that by combining the color information with the boundary shape of the object, the retrieval accuracy can be clearly improved. This can be done without increasing the dimensionality of the descriptor.

1 Introduction

Nowadays, the problem of image retrieval plays a remarkable role in the fields of image analysis and pattern recognition. With increasing amount of real-world image data to be processed and stored, the development of powerful retrieval tools has become an essential subject of research work. The description of the objects occurring in the images is based on visual features extracted from them. In addition to color and texture, shape is one of the most important features used to characterize the objects occurring in the images as accurately as possible. These features are widely used in content-based image retrieval systems [1],[8].

On the other hand, classification accuracy (effectiveness) of a certain descriptor is not an adequate measure for its usefulness in the retrieval. Due to the increasing number of online retrieval solutions, computational efficiency is nowadays considered equally important as effectiveness [10]. In retrieval applications, the matter of computational complexity is twofold, namely the cost of image database indexing and retrieval. In the indexing, the features (descriptors) are extracted from the database images. Although this part is not always online operation, the feature extraction should not be a computationally heavy. More importantly, retrieval is always performed in real time. Therefore, the descriptors used in retrieval are required to be compact. The compactness of a de-

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 415 – 422, 2005.

scriptor depends on its dimensionality, because the retrieval time is proportional to the descriptor dimensions. Consequently, low-dimensional descriptors are preferred.

In this paper, we concentrate on object description that is based on Fourier transform. Fourier-based methods are widely used in shape description [6]. Fourier descriptors have been found to be accurate in shape classification in several comparisons, [2],[3],[4],[9]. In addition to good retrieval and classification accuracy, there are also other reasons which make Fourier descriptors popular among the contour-based shape representations. The main advantages of the Fourier-based shape representations are that they are compact and computationally light methods. Furthermore, they are easy to normalize and their matching is very simple. Also the sensitivity to noise is low, when only low frequency Fourier coefficients are used as descriptors.

In addition to object shape, its color is often equally important feature. In retrieval systems, colors are usually characterized using relatively high-dimensional descriptors, like histograms [1] or other statistical measures. On the other hand, the number of descriptors that efficiently combine color and shape is very small. In the work of Mehtre et al., [5] color and shape of the object were combined. This approach, however, was based on quite complicated clustering method. Furthermore, the approach used moment-based shape features that are computationally more expensive than for example Fourier descriptors.

In this paper, we present a new approach to the use of Fourier descriptors in the characterization of image content. Hence, we show that the Fourier descriptor is capable of describing also other features of the object than its contour. In our approach, we add the object color to the Fourier-based contour description. In this way, the obtained descriptor is able to more accurate object description in the retrieval process. However, the color information does not increase the dimensionality of the obtained descriptor.

The organization of this paper is the following. In section two, the principles of Fourier descriptors and our approach, *Color Area Fourier*, are presented. Section three reports the retrieval experiments carried out using real industrial defect images. The proposed method is discussed in section four.

2 Object Description

In this section, the common methods for shape description using Fourier-based methods are presented. In addition to that, our approach to combine object color information with its contour in Fourier description is presented.

2.1 Fourier Descriptors

Shape signatures. Shape signature is a 1D boundary function $f(k)$ that represents the boundary of a 2D object. The functions are either real-valued or complex. Complex coordinate function [2] is the simplest and best-known boundary presentation. It presents the coordinates of the boundary (x_k, y_k) in an object centered complex coordinate system:

$$z(k) = (x_k - x_c) + j(y_k - y_c) \tag{1}$$

for k=0,1,2,…,N-1, in which N is the length of the boundary and (x_c, y_c) is the centroid of the object. Area function [11] is an example of real-valued shape signatures and it is defined as the area of the triangle formed by two boundary points and centroid in the object centered coordinate system:

$$a(k) = \frac{\left|(x_k - x_c)(y_{k+1} - y_c) - (x_{k+1} - x_c)(y_k - y_c)\right|}{2} \tag{2}$$

Hence, both signatures represent the boundary independent of the location of the object in the image.

Signatures for color and shape. The object signatures of the proposed descriptors use the same basic approach as the complex-valued shape signatures of equation (1). Hence, by combining two real-valued 1D signals it is possible to form a 1D complex signal. In this paper, we combine the object color to its boundary information. This is made by combining the color of the object region defined by the shape signature with the signature itself. The color value C_k can be e.g. the mean of the selected color component at each object region k. In the case of area function, the region corresponds to the image pixels covered by the area of the triangle defined by equation (2). The signature for color and shape of an object is expressed as complex numbers:

$$c_a(k) = a_k + jC_k \tag{3}$$

Hence the signature of equation (3) combines the real-valued boundary information with object color distribution.

Fourier description. The descriptor based on a signature function can be formed in several ways. Fourier transform is a commonly used method for this purpose. Fourier transformation of a boundary function generates a set of complex numbers which are called Fourier descriptors. Fourier descriptors characterize the object shape in a frequency domain. The discrete Fourier transform for a boundary function $f(k)$ is:

$$F_n = \frac{1}{N}\sum_{k=0}^{N-1} f(k)e^{-j2\pi nk/N} \tag{4}$$

for n=0,1,2,…,N-1. The general shape of the object is represented by the lower frequency descriptors, whereas high frequency descriptors represent the fine details of the object shape. The descriptors have to be normalized to achieve invariance to translation, rotation, and scaling. Translation invariance of is based on the object centered shape signatures. The descriptors can be made rotation invariant by ignoring the phase information and using only the magnitudes of the transform coefficients $|F_n|$. The scale can be normalized by dividing the magnitudes by $|F_0|$ or $|F_1|$, depending on the shape representation method [2].

Feature vectors. A common approach to object description is to use only a subset of low frequency coefficients that carry the most of the object information. This way the shape can be effectively presented using a relatively short feature vector. For complex-valued shape signatures, the coefficients are taken from positive and negative frequency axis:

$$x = \left[\frac{|F_{-(L/2-1)}|}{|F_1|}, \cdots, \frac{|F_{-1}|}{|F_1|}, \frac{|F_2|}{|F_1|}, \cdots, \frac{|F_{L/2}|}{|F_1|} \right]^T \tag{5}$$

in which L is a constant value that defines the dimensionality of the feature vector. When this description is formed for the transform coefficients obtained from complex coordinate function of equation (1), it is called *Contour Fourier* method [2]. In this paper, this kind of feature vector is applied also to *Color Area Fourier* descriptor that uses complex-valued signature of equation (3). However, in the case of *Color Area Fourier* descriptor, the normalization is carried out using $|F_0|$ instead of $|F_1|$:

$$x = \left[\frac{|F_{-(L/2)}|}{|F_0|}, \cdots, \frac{|F_{-1}|}{|F_0|}, \frac{|F_1|}{|F_0|}, \cdots, \frac{|F_{L/2}|}{|F_0|} \right]^T \tag{6}$$

The difference between the feature vectors can be explained by differences between the signatures. In the case of *Contour Fourier* method of equation (5), the signature uses merely boundary information that is represented in location-independent manner. Therefore, scale is normalized using the first non-zero coefficient, $|F_1|$. On the other hand, *Color Area Fourier* descriptor uses complex-valued signature of equation (3) in which contour shape is represented by centroid distance. Therefore, the mean value of the signature function differs from zero. This causes the normalization by $|F_0|$, which is the transform coefficient representing the mean value of the signal.

The real-valued shape representation, *Area Fourier* [2] uses the area function as shape signature. Because this signature is real, only half of the transform coefficients are needed to characterize the shape [2]:

$$x = \left[\frac{|F_1|}{|F_0|}, \frac{|F_2|}{|F_0|}, \cdots, \frac{|F_L|}{|F_0|} \right]^T \tag{7}$$

Also with this descriptor type, the normalization is carried out using the mean component, $|F_0|$, to remove the effect of the mean of the area function.

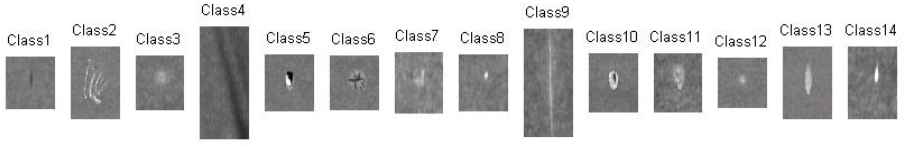


Fig. 1. An example image of each paper defect class in the testing database

3 Retrieval Experiments

In this section, we present the retrieval experiments carried out using a real defect image database. We compare the retrieval performance of the proposed *Color Area Fourier* approach to the ordinary Fourier shape description methods, which describe only the boundary line of the object.

3.1 Defect Image Database

For testing purposes, we used paper defect images that were collected from an industrial process. The images were taken from the paper manufacturing process using a paper inspection system [7] that produces gray level images of the defects. The reason for the collection of the defect image databases in the process industry is the practical need of controlling the quality and production [7]. In industrial imaging solutions, there is a need to retrieve the defect images from the databases. In these images, the defect shape and gray level are the most essential features that describe the defect type. Therefore, effective methods for the shape and gray level representation are needed in the retrieval and classification of the defect images. The defects occurring in the paper can be for example holes, wrinkles or different kinds of dirt spots. The test set consisted of 1204 paper defects, which represented 14 defect classes so that each class consisted of 27-103 images. An example image of each paper defect class is presented in figure 2. Within the classes of the defect database, there were differences in the gray levels, size, and orientation of the defects.

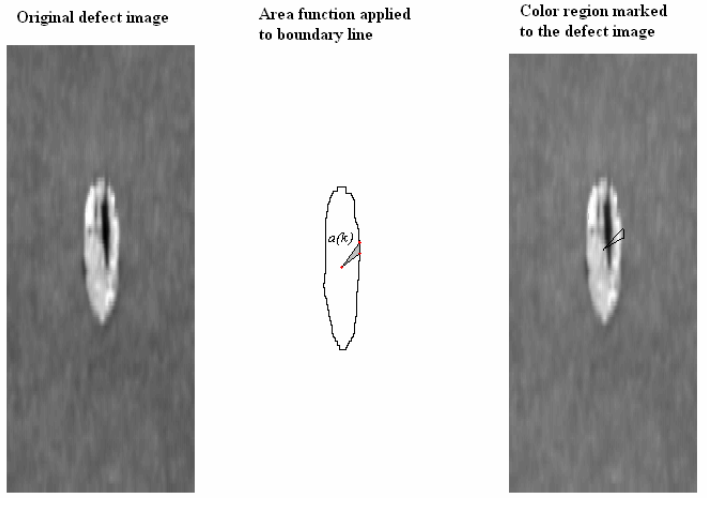


Fig. 2. The principle of *Color Area Fourier* descriptor applied to a paper defect image

3.2 Retrieval

In this paper, we compare our *Color Area Fourier* approach to *Area Fourier* method and *Contour Fourier* method that have been proved to be effective Fourier-based shape descriptors in retrieval of defect shapes [3].

The Fourier-based shape descriptors (*Area Fourier* and *Contour Fourier*) were calculated based on the defect contours. The descriptors used triangular color regions that were defined from the defect images so that the triangle was drawn between object centroid and two consequent boundary points (figure 2). The feature vectors of the descriptors were formed using equations (7) and (5), respectively. In the case of *Color Area Fourier* descriptor, the color information was added to the area-based shape description. The defects are presented as gray level images, which means that only the intensity component was used to represent the color. The color value C_k was selected to be the mean gray level of the triangular region. The descriptors were formed using feature vector of equation (6).

In the comparison, low-dimensional descriptors were preferred, and hence we used two lengths of the vectors (L), namely 8 and 16. In the retrieval experiments, the distance measure between the feature vectors was selected to be the Euclidean distance. The retrieval experiments were made using *leaving one out* method. In this method, each image in turn is left out from the test set and used as a query image; whereas the other images in the test set form a testing database. The performance of the retrieval was measured by calculating a *precision versus recall curve* for each query.

The average precision/recall curves for the database are presented in figure 3. The results show that the Fourier descriptors combined with object color (*Color Area Fourier*) outperform clearly the *Area Fourier* descriptor. On the other hand, the

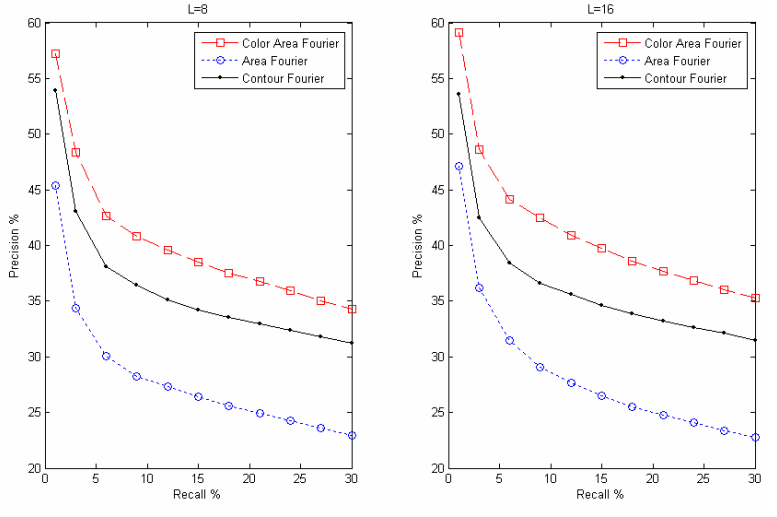


Fig. 3. The average precision/recall curves of the retrieval experiments calculated for each descriptor type

proposed descriptors outperform also *Contour Fourier* method that is the most accurate shape-based Fourier descriptor in defect image retrieval [3]. It is essential to note that this improvement does not increase dimensionality of the feature vectors, when *Color Area Fourier* and *Contour Fourier* descriptors are compared. Furthermore, when the same distance metrics is applied, the computational cost of retrieval is equal with the proposed descriptors and the conventional Fourier shape descriptors.

4 Discussion

In this paper, a new approach to Fourier-based object presentation was introduced. Our approach, *Color Area Fourier* descriptor, combines the color and shape information of an object into a single feature vector. The obtained vector is as low dimensional and easy to match as any other shape-based Fourier descriptor. However, our experiments showed that *Color Area Fourier* descriptor outperforms the other Fourier-based shape descriptors in terms of retrieval accuracy.

In the experiments, a database of complex shapes was used. The shapes in the database represent defects that occur in an industrial process. Self-evidently, the accuracy of the descriptors is the most important criterion also in this retrieval problem. On the other hand, the matter of computational efficiency is essential in this case, like in the most of the real-world image retrieval tasks. Therefore, compact features are required. The experiments showed that combining the color information to a Fourier descriptor, additional retrieval accuracy can be achieved without increasing computational cost of retrieval. Therefore, the *Color Area Fourier* descriptor presented in this paper is an effective and efficient tool for describing complex objects in image retrieval and classification.

References

1. Del Bimbo, A.: Visual Information Retrieval, Morgan Kaufmann Publishers, San Fransisco, (2001)
2. Kauppinen, H., Seppänen, T., Pietikäinen, M.: An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 2 (1995) 201-207
3. Kunttu, I., Lepistö, L., Rauhamaa, J., Visa, A.: Multiscale Fourier Descriptor for Shape-Based Image Retrieval, Proceedings of 17th International Conference on Pattern Recognition, Cambridge, UK, Vol. 2 (2004) 765-768
4. Mehtre, B.M., Kankanhalli, M.S., Lee, W.F.: Shape Measures for Content Based Image Retrieval: A Comparison, Information Processing Management, Vol. 33, No 3, (1997) 319-337
5. Mehtre, B.M., Kankanhalli, M.S., Lee, W.F.: Content-Based Image Retrieval Using a Composite Color-Shape Approach. Information Processing & Management Vol. 34, No. 1, (1998) 109-120.
6. Persoon, E., Fu, K.: Shape Discrimination Using Fourier Descriptors, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 7 (1977) 170-179

7. Rauhamaa, J., Reinius, R.: Paper Web Imaging with Advanced Defect Classification, Proceedings of the 2002 TAPPI Technology Summit, Atlanta, Georgia (2002)
8. Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-Based image Retrieval at the End of the Early Years. IEEE Transactions Pattern Analysis and Machine Intelligence Vol. 22, No. 12 (2000) 1349-1380.
9. Zhang, D., Lu, G.: A Comparative Study of Curvature Scale Space and Fourier Descriptors for Shape-Based image Retrieval, Journal of Visual Communication and Image Representation, Vol. 14, No. 1 (2003) 41-60
10. Zhang, D., Lu, G.: Review of shape representation and description techniques. Pattern Recognition, Vol. 37, No. 1, (2004) 1-19
11. Zhang, D.S., Lu, G.: Study and evaluation of different Fourier methods for image retrieval. Image and Vision Computing Vol. 23, (2005) 33-49

Face Recognition Using a Surface Normal Model

W.A.P. Smith and E.R. Hancock

Department of Computer Science, The University of York
{wsmith, erh}@cs.york.ac.uk

Abstract. This paper describes how facial shape can be modelled using a statistical model that captures variations in surface normal direction. We fit the model to intensity data using constraints on the surface normal direction provided by Lambert's law. We demonstrate that this process yields improved facial shape recovery and can be used for the purposes of illumination insensitive face recognition.

1 Introduction

Although alluring, the use of shape-from-shading for 3D reconstruction and recognition has proved to be an elusive task. This can mainly be attributed to the local convexity-concavity instability due to the bas-relief ambiguity. One way of overcoming this problem with single view shape-from-shading is to use domain specific constraints. Several authors [1,2] have shown that, at the expense of generality, the accuracy of recovered shape information can be greatly enhanced by restricting a shape-from-shading algorithm to a particular class of objects. Zhao and Chellappa [2] introduced a geometric constraint which exploited the approximate bilateral symmetry of faces. Atick et al. [1] proposed a statistical shape-from-shading framework based on a low dimensional parameterisation of facial surfaces.

However, it is surface orientation and not depth which is conveyed by image intensity. It is for this reason that in this paper we investigate whether surface orientation rather than depth can be used to construct a statistical model of face shape. Unfortunately, the construction of a statistical model for the distribution of facial needle-maps is not a straightforward task. The reason for this is that the statistical representation of directional data has proved to be considerably more difficult than that for Cartesian data. For instance, if we consider a short walk across one of the poles of the unit sphere, then although the distance traversed is small, the change in azimuth angle is large.

To overcome the problem, in this paper we draw on ideas from cartography. Our starting point is the *azimuthal equidistant* projection [3]. This projection has the important property that it preserves the distances between between the centre of projection and all other locations on the sphere. Another useful property of this projection is that straight lines on the projected plane through the centre of projection correspond to great circles on the sphere. We exploit these properties to generate a local representation of the field of surface normals. We commence with a set of needle-maps, i.e. fields of surface normals which

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 423–430, 2005.

in practice are obtained either from range images or shape-from-shading. We begin by computing the mean field of surface normals. The surface normals are represented using elevation and azimuth angles on a unit sphere. At each image location the mean-surface normal defines a reference direction which we use to construct an azimuthal equidistant projection for the distribution of surface normals at this point. The distribution of points on the projection plane preserves the distances of the surfaces normals on the unit sphere with respect to the mean surface normal, or reference direction. We then construct a deformable model over the set of surface normals by applying the Cootes and Taylor [4] point distribution model to the co-ordinates that result from transforming the surface normals from the unit sphere to the tangent plane under azimuthal equidistant projection.

The model provides a global statistical constraint which we exploit to help resolve the ambiguity in the shape-from-shading process. In addition, the model parameters describing a recovered needle-map are invariant to changes in illumination. We therefore use these parameters to perform illumination insensitive face recognition.

2 A Statistical Surface Normal Model

A "needle map" describes a surface $z(x,y)$ as a set of local surface normals $\mathbf{n}(x,y)$ projected onto the view plane. Let $\mathbf{n}_k(i,j) = (n_k^x(i,j), n_k^y(i,j), n_k^z(i,j))^T$ be the unit surface normal at the pixel indexed (i,j) in the k^{th} training image. If there are T images in the training set, then at the location (i,j) the mean-surface normal direction is $\hat{\mathbf{n}}(i,j) = \frac{\bar{\mathbf{n}}(i,j)}{||\bar{\mathbf{n}}(i,j)||}$ where $\bar{\mathbf{n}}(i,j) = \frac{1}{T}\sum_{k=1}^{T}\mathbf{n}_k(i,j)$.

On the unit sphere, the surface normal $\mathbf{n}_k(i,j)$ has elevation angle $\theta_k(i,j) = \frac{\pi}{2} - \arcsin n_k^z(i,j)$ and azimuth angle $\phi_k(i,j) = \arctan\frac{n_k^y(i,j)}{n_k^x(i,j)}$, while the mean surface normal at the location (i,j) has elevation angles $\hat{\theta}(i,j) = \frac{\pi}{2} - \arcsin\hat{n}^z(i,j)$ and azimuth angle $\hat{\phi}(i,j) = \arctan\frac{\hat{n}^y(i,j)}{\hat{n}^x(i,j)}$.

To construct the azimuthal equidistant projection we commence by constructing the tangent plane to the unit-sphere at the location corresponding to the mean-surface normal. We establish a local co-ordinate system on this tangent plane. The origin is at the point of contact between the tangent plane and the unit sphere. The x-axis is aligned parallel to the local circle of latitude on the unit-sphere. Under the azimuthal equidistant projection at the location (i,j), the surface normal $\mathbf{n}_k(i,j)$ maps to the point with co-ordinate vector $\mathbf{v}_k(i,j) = (x_k(i,j), y_k(i,j))^T$. The transformation equations between the unit-sphere and the tangent-plane co-ordinate systems are

$$x_k(i,j) = k' \cos\theta_k(i,j)\sin[\phi_k(i,j) - \hat{\phi}(i,j)]$$

$$y_k(i,j) = k'\left\{\cos\hat{\theta}(i,j)\sin\phi_k(i,j) - \sin\hat{\theta}(i,j)\cos\theta_k(i,j)\cos[\phi_k(i,j) - \hat{\phi}(i,j)]\right\}$$

where $\cos c = \sin\hat{\theta}(i,j)\sin\theta_k(i,j) + \cos\hat{\theta}(i,j)\cos\theta_k(i,j)\cos[\phi_k(i,j) - \hat{\phi}(i,j)]$ and $k' = \frac{c}{\sin c}$.

The equations for the inverse transformation from the tangent plane to the unit-sphere are

$$\theta_k(i,j) = \sin^{-1}\left\{\cos c \sin\hat{\theta}(i,j) - \frac{1}{c}y_k(i,j)\sin c\cos\hat{\theta}(i,j)\right\}$$
$$\phi_k(i,j) = \hat{\phi}(i,j) + \tan^{-1}\psi(i,j)$$

where

$$\psi(i,j) = \begin{cases} \frac{x_k(i,j)\sin c}{c\cos\hat{\theta}(i,j)\cos c - y_k(i,j)\sin\hat{\theta}(i,j)\sin c} & \text{if } \hat{\theta}(i,j) \neq \pm\frac{\pi}{2} \\ -\frac{x_k(i,j)}{y_k(i,j)} & \text{if } \hat{\theta}(i,j) = \frac{\pi}{2} \\ \frac{x_k(i,j)}{y_k(i,j)} & \text{if } \hat{\theta}(i,j) = -\frac{\pi}{2} \end{cases}$$

and $c = \sqrt{x_k(i,j)^2 + y_k(i,j)^2}$.

For each image location the transformed surface normals from the T different training images are concatenated and stacked to form two long-vectors of length T. For the pixel location indexed (i,j), the first of these is the long vector with the transformed x-co-ordinates from the T training images as components, i.e. $\mathbf{V}_x(i,j) = (x_1(i,j), x_2(i,j), ..., x_T(i,j))^T$ and the second long-vector has the y coordinate as its components, i.e. $\mathbf{V}_y(i,j) = (y_1(i,j), y_2(i,j), ..., y_T(i,j))^T$. Since the equidistant azimuthal projection involves centering the local co-ordinate system, the mean long-vectors over the training images are zero. If the data is of dimensions M rows and N columns, then there are $M \times N$ pairs of such long-vectors. The long vectors are ordered according to the raster scan (left-to-right and top-to-bottom) and are used as the columns of the $T \times (2MN)$ data-matrix $\mathbf{D} = (\mathbf{V}_x(1,1)|\mathbf{V}_y(1,1)|\ \mathbf{V}_x(1,2)|\mathbf{V}_y(1,2)|\ \ldots|\mathbf{V}_x(M,N)|\mathbf{V}_y(M,N))$. The covariance matrix for the long-vectors is the $(2MN)\times(2MN)$ matrix $\mathbf{L} = \frac{1}{T}\mathbf{D}^T\mathbf{D}$. We follow Atick et al. [1] and use the numerically efficient method of Sirovich [5] to compute the eigenvectors $\mathbf{e}_i$ of $\mathbf{L}$. We deform the equidistant azimuthal point projections in the directions defined by the $2MN \times K$ matrix $\mathbf{P} = (\mathbf{e}_1|\mathbf{e}_2|\ldots|\mathbf{e}_K)$ formed from the leading K principal eigenvectors. This deformation displaces the transformed surface normals on the local tangent planes in the directions defined by the eigenvectors $\mathbf{P}$. If $\mathbf{b} = (b_1, b_2, ..., b_K)^T$ is a vector of parameters of length K, then since the mean-vector of co-ordinates resulting from the equidistant azimuthal projection is zero, the deformed vector of projected co-ordinates is $\mathbf{v}_b = \mathbf{P}\mathbf{b}$. Suppose that $\mathbf{v}_o$ is the vector of co-ordinates obtained by performing the azimuthal equidistant projection on an observed field of surface normals. We seek the parameter vector $\mathbf{b}$ that minimises the squared error $\mathcal{E}(\mathbf{b}) = (\mathbf{v}_o - \mathbf{P}^T\mathbf{b})^T(\mathbf{v}_o - \mathbf{P}^T\mathbf{b})$. The solution to this least-squares estimation problem is $\mathbf{b}^* = \mathbf{P}^T\mathbf{v}_o$. The best fit field of surface normals allowed by the model is $\mathbf{v}_o^* = \mathbf{P}\mathbf{P}^T\mathbf{v}_o$. The deformed vector of azimuthal equidistant projection co-ordinates can be transformed back into a surface normal on the unit sphere using the inverse azimuthal equidistant projection equations given above.

3 Fitting the Model to Intensity Images

We may exploit the statistical constraint provided by the model in the process of fitting the model to an intensity image and thus help resolve the ambiguity in the shape-from-shading process. We do this using an iterative approach which can be posed as that of recovering the best-fit field of surface normals from the statistical model, subject to constraints provided by the image irradiance equation.

If I is the measured image brightness, then according to Lambert's law $I = \mathbf{n}.\mathbf{s}$, where $\mathbf{s}$ is the light source direction. In general, the surface normal $\mathbf{n}$ can not be recovered from a single brightness measurement since it has two degrees of freedom corresponding to the elevation and azimuth angles on the unit sphere. In the Worthington and Hancock [6] iterative shape-from-shading framework, data-closeness is ensured by constraining the recovered surface normal to lie on the reflectance cone whose axis is aligned with the light-source vector $\mathbf{s}$ and whose opening angle is $\alpha = \arccos I$. At each iteration the surface normal is free to move to an off-cone position subject to smoothness or curvature consistency constraints. However, the hard irradiance constraint is re-imposed by rotating each surface normal back to its closest on-cone position. This process ensures that the recovered field of surface normals satisfies the image irradiance equation after every iteration.

Suppose that $\mathbf{n'}^l(i,j)$ is an off-cone surface normal at iteration l of the algorithm. The update equation is therefore $\mathbf{n}^{l+1}(i,j) = \Theta\mathbf{n'}^l(i,j)$ where Θ is a rotation matrix computed from the apex angle α and the angle between $\mathbf{n'}^l(i,j)$ and the light source direction $\mathbf{s}$. To restore the surface normal to the closest on-cone position it must be rotated by an angle $\theta = \alpha - \arccos\left[\mathbf{n'}^l(i,j).\mathbf{s}\right]$ about the axis $(u,v,w)^T = \mathbf{n'}^l(i,j) \times \mathbf{s}$. Hence, the rotation matrix is

$$\Theta = \begin{pmatrix} c + u^2c' & -ws + uvc' & vs + uwc' \\ ws + uvc' & c + v^2c' & -us + vwc' \\ -vs + uwc' & us + vwc' & c + w^2c' \end{pmatrix}$$

where $c = cos(\theta)$, $c' = 1 - c$ and $s = \sin(\theta)$.

The framework is initialised by placing the surface normals on their reflectance cones such that they are aligned in the direction opposite to that of the local image gradient (biasing towards global convexity).

Our approach to fitting the model to intensity images uses the fields of surface normals estimated using the geometric shape-from-shading method described above. This is an iterative process in which we interleave the process of fitting the statistical model to the current field of estimated surface normals, and then re-enforcing the data-closeness constraint provided by Lambert's law by mapping the surface normals back onto their reflectance cones. The algorithm can be summarised as follows:

1. Calculate an initial estimate of the field of surface normals $\mathbf{n}$ by aligning each normal on its reflectance cone with the negative local intensity gradient.

2. Each normal in the estimated field $\mathbf{n}$ undergoes an azimuthal equidistant projection to give a vector of transformed coordinates $\mathbf{v}_o$.
3. The vector of best fit model parameters is given by $\mathbf{b} = \mathbf{P}^T\mathbf{v}_o$.
4. The vector of transformed coordinates corresponding to the best-fit parameters is given by $\mathbf{v}' = \mathbf{P}\mathbf{P}^T\mathbf{v}_o$.
5. Using the inverse azimuthal equidistant projection find the off-cone best fit surface normal $\mathbf{n}'$ from $\mathbf{v}'$.
6. Find the on-cone surface normal $\mathbf{n}''$ by rotating the off-cone surface normal $\mathbf{n}'$ using $\mathbf{n}'' = \mathbf{\Theta}\mathbf{n}'$.
7. Test for convergence. If $\sum_{i,j} \cos^{-1}\left[\mathbf{n}(i,j).\mathbf{n}''(i,j)\right] < \epsilon$, where ϵ is a predetermined threshold, then stop and return $\mathbf{b}$ as the estimated model parameters and $\mathbf{n}''$ as the recovered needle map.
8. Make $\mathbf{n} = \mathbf{n}''$ and return to step 2.

Since real world face images contain albedo variations, we choose to output $\mathbf{n}'$ and estimate the facial albedo map using the differences between observed and reconstructed image brightness, i.e. we relax the data-closeness constraint at the final iteration. Hence, the albedo ρ is given by $\rho(i,j) = \frac{I(i,j)}{\mathbf{s}.\mathbf{n}'(i,j)}$, where $\mathbf{s}$ is the light source vector.

4 Experiments

In Figure 1 we illustrate the results of the model fitting process. We train the statistical model using surface normals extracted from 200 range images of male and female subjects in frontal poses and neutral expressions [7]. We fit the statistical model to an image using the technique described in Section 3. As input we use images of 10 subjects from the Yale B database [8] in frontal pose and illuminated by a single light source with direction $[0\ 0\ 1]^T$. The algorithm typically converged within 20 iterations. We show the surfaces recovered by integrating the best fit needle maps using the technique of Frankot and Chellappa [9]. In the first and third rows the surfaces are shown rotated 30° about the vertical axis. The surfaces are rendered with Lambertian reflectance and the estimated albedo maps. The light source remains fronto-parallel with respect to the face. The resulting synthesised images are near photo-realistic under a large change in viewpoint. Certainly, the results are comparable with those of Georghiades et al. [8] in which 7 input images were required per subject. The second and fourth rows of Figure 1 show the meshes of the recovered surfaces to allow inspection of the recovered shape alone. In Figure 2 we demonstrate that the recovered surface and albedo map are sufficiently stable to synthesise images in both novel pose and novel illumination. We show the surface of the eighth subject from the previous figure and circle the light source from left profile to right profile.

Provided that the shape-from-shading process is sufficiently accurate, the parameters describing a recovered facial needle map are invariant to illumination and reflectance properties. They hence encode only appearance. For this reason these parameters potentially provide a means of performing illumination and reflectance invariant face recognition. In our experiments we use a

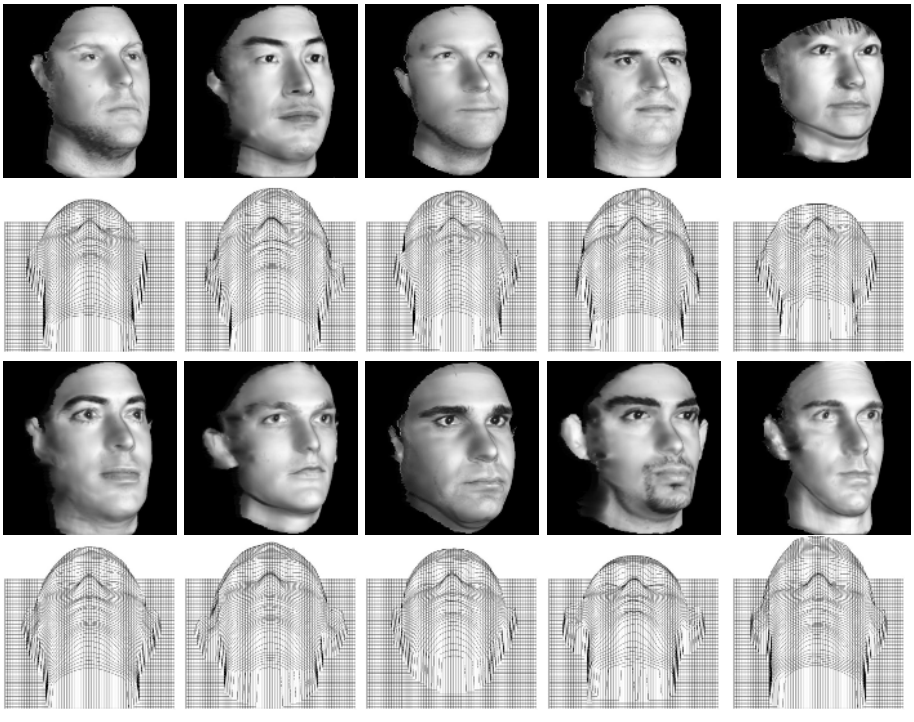


Fig. 1. Surfaces recovered from the ten subjects in the Yale B database

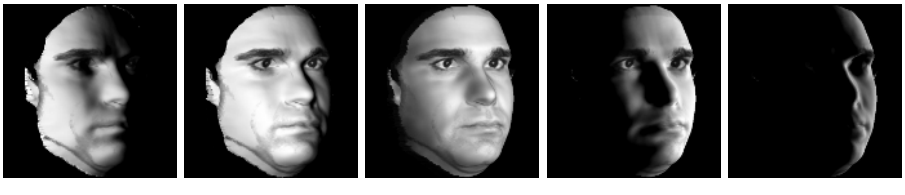


Fig. 2. Surface recovered from subject 8 in novel pose and under varying illumination

subset of the CMU PIE database [10]. This database contains images of 67 subjects under varying pose, illumination and expression. We fix the pose to fronto-parallel (camera c27) and vary the illumination direction along a horizontal arc approximately 55° in each direction. The set of flashes used was {f03,f10,f07,f08,f09,f13,f16}. For each subject we use only one gallery image, illuminated from close to the viewing direction (flash f08). We fit the statistical model to each gallery image which provides an appearance vector for each subject. For each probe image we repeat the same process to find an appearance vector for the unknown face.

In order to effect recognition with 1 gallery image per subject, we find the Euclidian distances between a given probe vector and all the gallery vectors and sort them. A probe has rank k if the correct match is the k^{th} smallest Euclidian

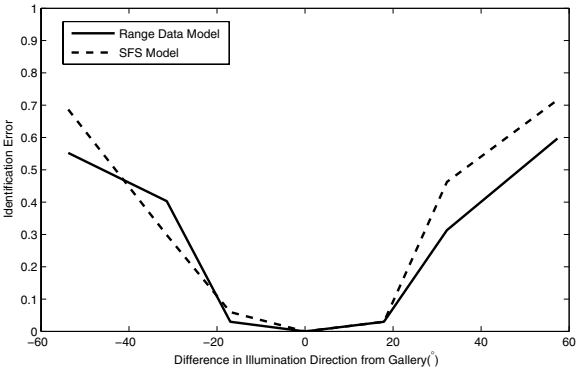
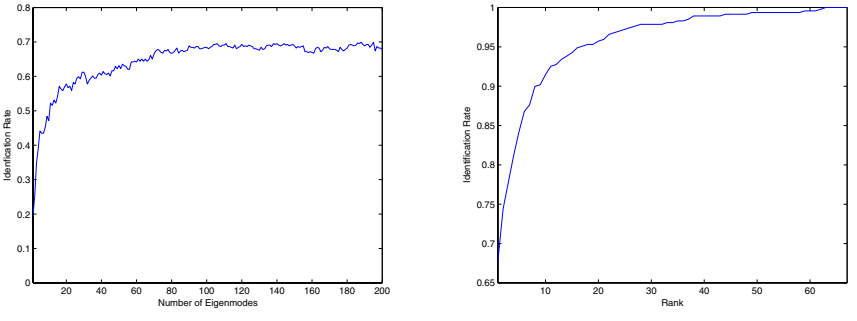


Fig. 3. Recognition error versus angle of illumination



(a) (b)

Fig. 4. Identification rate versus (a) the number of Eigenmodes used and (b) rank

distance. A probe is correctly identified if it has rank 1, i.e. the correct match has the lowest Euclidian distance. The rank can vary between 1 and the size of the gallery (i.e. 67). In Figure 3 we begin by showing the rank 1 identification error as the illumination angle is varied through approximately 110°. We experiment with both the model trained on ground truth normals extracted from range data and on normals extracted using shape-from-shading.

From the plot it is clear that low error rates are achievable for variations in illumination direction of approximately ±30°. Beyond this point, performance decreases rapidly. We believe this is because as the illumination direction becomes more extreme, increasing areas of the face are in shadow. Thus, the imposition of data-closeness at each iteration in these shadow areas may result in the fitting process 'walking away' from the true solution. Under these conditions, the iterative fitting process would be best placed in a statistical setting which attempts to match the model to the needle map in the visible areas only.

In Figure 4 (a) we investigate how the number of eigenmodes used affects identification performance. We show the total rank 1 identification rate across all illumination conditions against the number of eigenmodes used. Performance

appears to level out at around 100 eigenmodes (dimensions), suggesting that additional dimensions do not encode modes of facial shape which are useful for recognition.

The identification rate at rank k is the fraction of probes that have rank k or lower. Plotting identification performance against rank on a cumulative match characteristic (CMC) allows us to asses whether the correct match is in the top k matches rather than just the first. Most applications are based on performance at lower ranks. From the CMC in Figure 4 (b) it can be seen that performance rises sharply from rank 1 to 10.

5 Conclusions

We have shown how a statistical model of shape may be constructed from fields of surface normals using the azimuthal equidistant projection. We presented an iterative method for fitting the model to an image subject to image irradiance constraints. The method proves rapid to converge, and delivers realistic surfaces when the fields of surface normals are integrated. The resulting parameter vector provides a means to perform illumination insensitive face recognition. The technique could also be use as a generative model to sample the entire pose and illumination space from a single image. Our future plans revolve around placing the iterative process in a statistical setting in order to improve the reliability of the estimated parameters under varying illumination. We also plan to develop ways of aligning the model with images which are not in a frontal pose.

References

1. Atick, J.J., Griffin, P.A., Redlich, A.N.: Statistical approach to SFS: Reconstruction of 3D face surfaces from single 2D images. Neural Comp. **8** (1996) 1321–1340
2. Zhao, W.Y., Chellappa, R.: Illumination-insensitive face recognition using symmetric SFS. In: Proc. CVPR. (2000)
3. Snyder, J.P.: Map Projections–A Working Manual, U.S.G.S. Professional Paper 1395. United States Government Printing Office, Washington D.C. (1987)
4. Cootes, T.F., J.Taylor, C., Cooper, D., Graham, J.: Training models of shape from sets of examples. In: Proc. BMVC. (1992) 9–18
5. Sirovich, L.: Turbulence and the dynamics of coherent structures. Quart. Applied Mathematics **XLV** (1987) 561–590
6. Worthington, P.L., Hancock, E.R.: New constraints on data-closeness and needle map consistency for shape-from-shading. IEEE Trans. PAMI **21** (1999) 1250–1267
7. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Computer Graphics Proc. SIGGRAPH. (1999) 187–194
8. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. PAMI **23** (2001) 643–660
9. Frankot, R.T., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. IEEE Trans. PAMI **10** (1988) 439–451
10. Sim, T., Baker, S., Bsat, M.: The cmu pose, illumination, and expression database. IEEE Trans. Pattern Analysis and Machine Intelligence **25** (2003) 1615–1618

A Robust Two Stage Approach for Eye Detection

Jing-Wein Wang and Chin-Chun Kuo

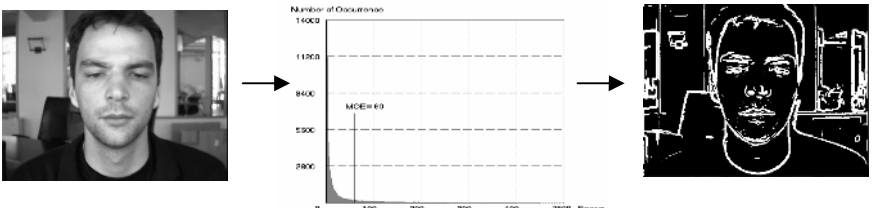
Institute of Photonics and Communications,
National Kaohsiung University of Applied Sciences,
415 Chien-Kung Road, Kaohsiung, Taiwan 807, R.O.C.
Tel. +886-7-3814526 Ext. 3350
Fax. +886-7-3832771
jwwang@cc.kuas.edu.tw

Abstract. This paper adopts face localization to eye extraction strategy for eye detection in complex scenes. First, an energy analysis is applied to enhance face localization performance by removing most noise-like regions rapidly. According to anthropometry, the face-of-interest (FOI) region is located using signatures derived from the proposed head contour detection (HCD) approach that searches the best combinations of facial sides and head contours. Second, via the de-edging preprocessing for facial sides, a wavelet subband inter-orientation projection method is devised to generate and select eye-like candidates. By utilizing the geometric discrimination information among the facial components, such as the eyes, nose, and mouth, the proposed eye verification rules verify the eye pair selected from the candidates. The experimental results demonstrate the significance performance improvement using the proposed method over others on three head-and-shoulder databases.

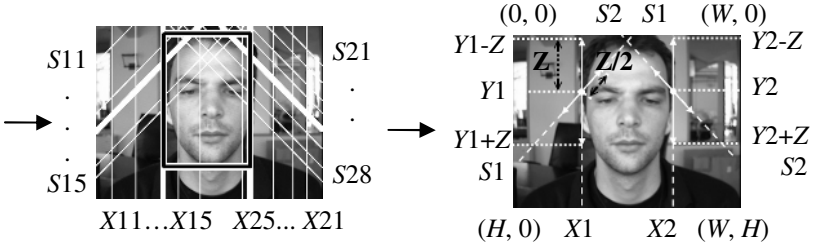
1 Introduction

Biometric technology such as eye detection in an image is a challenging problem because it involves locating eye with no prior knowledge about image content [1]. In this work, we propose the use of complementary techniques which are based on head contour geometry characterization and wavelet subband inter-orientation projection. The technique aims at providing an efficient system to operate with complex backgrounds and must tolerate illumination changes, scale variations, and small head rotations, say 30°. The presented eye detection framework composed of the face localization and eye extraction stages as shown in Fig. 1. This paper is organized as follows. The next section is dedicated to HCD approach being as a justified way of locating FOI region. Using FOI size estimate, we define a wavelet subband inter-orientation projection method for generating and selecting the eye-like candidates in Section 3. Eye extraction is achieved by examining the correlation between eyes and detecting geometrical relationships among the facial components such as the eyes, nose, and mouth. Finally, Section 4 gives the experimental results on three head-and-shoulder image databases and Section 5 concludes the paper.

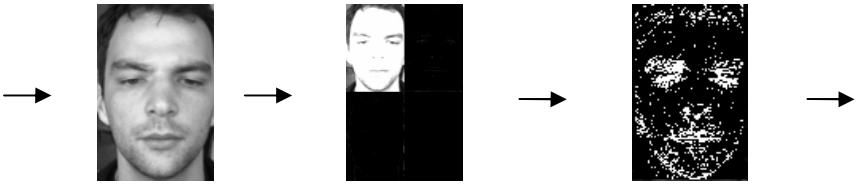
F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 431 – 437, 2005.



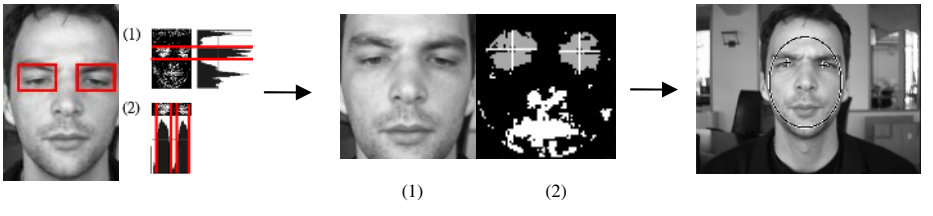
(a) Input image, (b) Histogram and MOE value of the energy image, (c) Result of energy thresholding,



(d) Head contour detection, (e) Edge confirmation criterion,



(f) Located FOI region, (g) One-level wavelet decomposition, (h) LH subband binary thresholding along with HL subband de-edging for facial sides,



(i) Candidate eye-pair generation and selection by using wavelet subband inter-orientation projection, (j) Both correlation coefficient and facial geometry relationship used for eye verification, (k) Output image.

Fig. 1. Block diagram of eye detection steps

2 Face Localization

Prior to face object extraction, a 3 × 3 smoothing filter is used to move the center from pixel to pixel in an image to guarantee the desired local edge enhancement. This

continues until all pixel locations have been covered and a new energy image is to be created for storing the response of the linear mask. Fig. 1(a) shows the example image of size 384 × 286 pixels with 256 gray levels, and Fig. 1(b) shows its energy histogram. To extract the interested object from the background, the threshold is set as mean of energy (MOE) initially and gradually changed until the observed pixel-of-thresholding (point) density of the whole image is around 10% ~ 20% which is considerably good representation for face to be detected as shown in Fig. 1(c) denoted as $B(x, y)$. Human head contour, which contains relatively concentrated information in face image, is more reliable than the feature of eyes or other face organs especially detecting unclear images or small face images. Locating eye with feature from head contour has the advantage of scale invariance and simplification. The algorithm consists of the following details as shown in Fig. 1(d). Below, we show the details of the proposed HCD algorithm, how effectively locates the head-face profile. Beginning, if

$$Nr(B(x,y)) = \begin{cases} 1, & \text{if } B(x,y) = 255 \\ 0, & \text{if } B(x,y) = 0 \end{cases}, \tag{1}$$

and $0 \le k \le H$, for each left diagonal projection, D_{lk} is computed by

$$D_{lk} = \sum_{(x,y)=(0,k)}^{(k,0)} Nr(B(x,y)), \tag{2}$$

and each left vertical projection, V_{lk} is computed by

$$V_{lk} = \sum_{(x,y)=(0,0)}^{(k,H)} Nr(B(x,y)). \tag{3}$$

Next, for each right diagonal projection, D_{rk} is computed by

$$D_{rk} = \sum_{(x,y)=(W,k)}^{(W-k,0)} Nr(B(x,y)), \tag{4}$$

and each right vertical projection, V_{rk} is computed by

$$V_{rk} = \sum_{(x,y)=(W,0)}^{(W-k,H)} Nr(B(x,y)). \tag{5}$$

According to the point projection waveforms from equations (2)-(5), the diagonal head boundary candidates and horizontal facial side candidates are determined from the peak responses with wide enough spreading which is defined as projection relief slope greater than 1.0, respectively. A pseudo line drawn from the valley to the peak of response measures the slope. Afterwards, the lines marked by white solid lines denote all candidate locations of head-face as presented in Fig. 1(d), i.e.

$$S1 \in \{S11, S12, S13, S14, S15\}, \tag{6}$$

$$S2 \in \{S21, S22, S23, ..., S28\}, \tag{7}$$

$$X1 \in \{X11, X12, X13, X14, X15\}, \tag{8}$$

$$X2 \in \{X21, X22, X23, X24, X25\}. \tag{9}$$

For each pair of head candidates $S1\times$ and $S2\times$, the algorithm gives the cost for each pair of facial side candidates $X1\times$ and $X2\times$, where $\times$ stands for a digit 1, 2, 3,... Next, the proposed edge confirmation criterion measures the fit of line candidates to the input image. We cite one illustrated example for localization as displayed in lines $S1$, $S2$, $X1$, and $X2$ of Fig. 1(e). The cost of the possible head-face boundary, lines $S1$ and $X1$, on the left side of the image is given by

$$\#LV1 = \sum_{i=Y1-Z}^{Y1-1} Nr(B(x,i)), \tag{10}$$

$$\#LV2 = \sum_{i=Y1}^{Y1+Z-1} Nr(B(x,i)), \tag{11}$$

$$\#LD1 = \sum_{i=0}^{\frac{Z}{2}-1} Nr(B(X1-i,Y1+i)), \tag{12}$$

$$\#LD2 = \sum_{i=0}^{\frac{Z}{2}-1} Nr(B(X1+i,Y1-i)), \tag{13}$$

$$\#LE = \#LV2 + \#LD2, \tag{14}$$

$$Nr(B(x,y)) = \begin{cases} 1 & if\ B(x,y) = 255 \\ 0 & if\ B(x,y) = 0 \end{cases}, \tag{15}$$

where $\#LV1$ and $\#LV2$ are number of points on the upper and lower line segments of $X1$, $\#LD1$ and $\#LD2$ are number of points on the lower and upper line segments of $S1$, and $\#LE$ is the sum of the line segments $\#LV2$ and $\#LD2$. Similarly, the cost of the possible head-face boundary, lines $S2$ and $X2$, on the right side of the image are given by $\#RV1$, $\#RV2$, $\#RD1$, $\#RD2$, and $\#RE$. While the cost of each boundary candidate measures the boundary possibility of each candidate, the filtering condition evaluated by using the equation (16) screens the candidate FOI regions and the decision condition made on both sides is based on the largest numbers #*LE* and #*RE* to complete detecting which the face locates, respectively.

$$\begin{aligned} &(\#LV2 > \#LV1)\ \&\&\ (\#LD2 > \#LD1)\ \&\& \\ &(\#RV2 > \#RV1)\ \&\&\ (\#RD2 > \#RD1)\ \&\& \\ &(\#LV2 > Z/4)\ \&\&\ (\#LD2 > Z/8)\ \&\& \\ &(\#RV2 > Z/4)\ \&\&\ (\#RD2 > Z/8), \end{aligned} \tag{16}$$

where Z denotes the region of check points for computation and we set Z to a value $H/4$. The result of Fig. 1(f) shows that a successful face localization is obtained based on the aspect ratio of the face shape, which has been set to be 6:4 in this algorithm. Once the HCD approach fails to detection, in what follows, the face localization will renew and search one of nine subimages with size two-thirds of the input image, which were given by overlap subdivision. The localization will not stop until an eye pair in the input image is found or the scans are over all the subimages.

3 Eye Extraction

The algorithm for the extraction and verification of the eye pair is described as follows:

Step 1: Perform discrete wavelet transforms (DWT) using D4 scalar wavelet for the FOI subimage as shown in Fig. 1(f) and take both the LH detail subband D_{LH} and HL detail subbad D_{HL} (Fig. 1(g)), since the separable sampling in DWT provides divisions of spectrum with sensitivity to horizontal eye edges and vertical facial sides, respectively.

Step 2: Remove the associated facial edge on the LH subband to avoid a false alarm in eyes projection, provided there is a significant peak response on either side of the HL subband.

Step 3: Binarize the FOI subimage obtained from the previous step to delete noise-like coefficients by adaptively selecting a reasonable threshold using the wavelet histogram of this region (Fig. 1(h)). Let B_{LH} represent the thresholded output.

Step 4: Project *y*-axis integrally and pick at most the first three peak responses as bases. The point projections of $B_{LH}(i, j)$ along its rows is given by

$$P_h(i) = \sum_{j=1}^{3n/4} B_{LH}(i, j) \;\; 1 \le i \le m \,. \tag{16}$$

The *y*-axis projection area is within the confines of three fourths $R^{m \times n}$ to avoid the possible false alarm caused by the mouth region. Then we can use *j* that has a maximum value of $\{P_h(j)\}_{\max}$ from $P_h(j)$ to restrict the *x*-axis projection region. We note that *j* in $\{P_h(j)\}_{\max}$ slightly varies according to the appearance within the candidate as shown in (1) of Fig. 1(i). Therefore, as shown in (2) of Fig. 1(i), we can decide on the basis of the coordinate $\{P_h(j)\}_{\max}$ to constraint the *x*-axis projection around the area delimited by the projective upper and lower valleys, $P_h(j)\}_{u\_\min}$ and $\{P_h(j)\}_{l\_\min}$, respectively. So the point projections of $B_{LH}(i, j)$ along its column is given by

$$P_v(i) = \sum_{i=1}^{m} B_{LH}(i, j) \;\; \{P_h(j)\}_{u\_\min} \le j \le \{P_h(j)\}_{l\_\min} \tag{17}$$

The above-presented inter-orientation projection does capture characteristics of the vertical profile and the horizontal symmetric distribution of human eye. Symmetrical projection relieves will encourage eye candidates to be identified, while asymmetrical projection relieves will favor others. According to anthropometry, we could circumscribe a circle area on the eye-mouth triangle by cropping the face of Fig. 1(c) for further screening the primary selection. For successful extraction of eyes, the eye-mouth circumscribed circle area (Fig. 1(j)) must pass over all examinations for verification. Based on facial component geometry eye-pair verification rules including eyes matching by correlation and mouth checking, which serves to tell us how like to each other they are, are adopted to verify the extracted eyes. The details are described in Step 5 below.

Step 5: The requirements in the eye verification procedure consist of the following:

1). The points of eye region are more than the ones of cheek region ((1) of Fig. 1(j)).
2). The points of both nose and mouth regions are more than the ones of two cheeks, or the points of mouth region are more than the ones of nose region ((1) of Fig. 1(j)).
3). The correlation coefficient is calculated to match the eye pair ((2) of Fig. 1(j)). Before obtaining the matching score between eyes, it is necessary to obtain normalizing for both size and rotation, which involves spatial scaling and finding of the eye pair centroids. Search the centroid in the eye region, we approach region segmentation by finding meaningful boundary based on point aggregation procedure. Choosing the center pixel of the eye region is a natural starting point and grouping points to form the region of interest with paying attention to 4-connectivity would yield a segmentation result, when no more pixels for inclusion in the region. The region segmentation result in general does not only give the required eye region, but also eyebrow or glasses, since there are still some thresholded wavelet coefficients located within the growing path between eye socket and eyebrow. In other words, besides the true eye landmarks, eyebrow and glasses, if any, will be included as well due to projection response spreading. After growing, the region centroid is relocated. By overlapping two centroids and with the help of two region borders we simply translate and rotate two regions so that they align themselves. The eye pattern decision is to perform matching via the correlation. The correlation factor was empirically determined as 0.5. The higher correlation value indicates that the two regions shapes are the best match. In determining the potential eye candidate using correlation matching, we aim to minimize the false acceptance rate with zero false rejection one. After the eyes are detected as described, the actual nose and the mouth locations, and the face size are also known.

Step 6: Refine the size and position of the FOI bounding box as an ellipse with the "+" symbol representing hitting the bull's-eye that is the most eyeball location. In order to make a compensation for region growing in the previous step, the "+" coordinates were modified by shifting one-third diameter of the grown region down in the vertical direction from the detected centroid coordinates. If the algorithm is failure to extract eyes, on the other hand, then it just outputs the earlier result of face localization (Fig. 1(k)).

4 Experimental Results

In order to evaluate the performance of the proposed system, we report on three different head-and-shoulder image sets, which are BioID [3], JAFFE [4], and Caltech [5]. The successful eyes detection was defined as excluding the rest face regions from the eye sockets. To make the results compared with the related works on the BioID and JAFFE test sets, we adopt the relatively error measure introduced by Jesorsky et al. [6]. The detection rate is hereby defined as the ratio of the number of correct detections to that of all images in the gallery. In Table 1, three test sets have been evaluated the performance of our method. From the experimental results on BioID, our system outperforms the results achieved by [6], [7], and [8]. On the JAFFE test set, our method compares favorably with the result reported in [7]. On the other hand, more false positives (erroneous) and false negatives (dismissal) than the other two sets are found on the Caltech test set, which has a little lower correct rate than the other two sets. For an overall evaluation, the experimental results show that the proposed eye detector can handle frontal view, facial variations (e.g., eye close or mouth open), pose, scale, and

view-independent rotation present in the database quite well. The correct rates for detecting eyes are all over 90%. However, the proposed system reports unsuccessful detection including false and missed eyes. We observe that our detector failed mainly for faces of too dark. If the facial components are not found because of lighting effect, which complicate the face localization task considerably, it is hardly to further extract eyes even though the face can be located. In terms of speed, the execution time of the presented detector is directly related to size and complexity of the images. For example, our system is operating at an average processing time 2.0 sec per BioID image (384 × 286 pixels) on a 1.4 GHz Pentium PC.

Table 1. Eye detection rates on three datasets using the HCD algorithm

Dataset (Sample number)	False positive	False negative	Correct
BioID (1521)	0.79% (12)	0.98% (15)	98.23% (1494)
JAFFE (213)	0% (0)	0.94% (2)	99.06% (211)
Caltech (450)	1.33% (6)	8.67% (39)	90.00% (405)

5 Conclusions

Experiments have shown high detection rates with a low number of false alarms on three datasets. To conclude, the proposed system demonstrates its robustness and high accuracy to that of the related works.

Acknowledgements

The financial support provided by the NSC 94-2622-E-151-014-CC3 is gratefully acknowledged.

References

[1] Jain, A. K., Ross, A., and Prabhakar, S.: An introduction to biometric recognition. IEEE Trans. Circuits and Systems for Video Technology, 14 (2004) 4-20.
[2] Mallat, S. (ed.): A Wavelet Tour of Signal Processing. Academic Press (1999).
[3] http://www.humanscan.de/
[4] http://www.mis.atr.co.jp/~mlyons/jaffe.html
[5] http://www.vision.caltech.edu/html-files/archive.html
[6] Kirchberg, K. J., Jesorsky, O., and Frischholz, R. W.: Genetic model optimization for Hausdorff distance-based face localization. Proc. ECCV2002, Denmark, LNCS-2359 (2002) 103-111.
[7] Zhou Z.-H. and Geng, X.: Projection functions for eye detection. Pattern Recognition, 37 (2004) 1049-1056.
[8] Wu, J. and Zhou, Z.-H.: Efficient face candidates selector for face detection. Pattern Recognition, 36 (2003) 1175-1186.

An Approximation of the Maximal Inscribed Convex Set of a Digital Object

Gunilla Borgefors[1] and Robin Strand[2]

[1] Centre for Image Analysis, Swedish University of Agricultural Sciences,
[2] Centre for Image Analysis, Uppsala University,
Lägerhyddsvägen 3, SE-75237 Uppsala, Sweden
{gunilla, robin}@cb.uu.se

Abstract. In several application projects we have discovered the need of computing the maximal inscribed convex set of a digital shape. Here we present an algorithm for computing a reasonable approximation of this set, that can be used in both 2D and 3D. The main idea is to iteratively identify the deepest concavity and then remove it by cutting off as little as possible of the shape. We show results using both synthetic and real examples.

1 Introduction

Shape manipulation and description in 2D and 3D are of great importance when analyzing many kinds of digital images. Many shape descriptors have been introduced, to capture various aspects of shape. Global descriptors, e.g., bounding box and compactness (P2A), are generally useful, as they make comparisons between objects easier than a set of local descriptors. However, they can be difficult to compute, especially in 3D images.

A popular tool for shape description is the *convex hull* (CH), i.e. the minimal area convex set that includes an object [1]. A concept that should be equally useful, but is much harder to compute, is the *maximal area/volume convex subset* (MACS), i.e. the largest convex set enclosed in the object. Finding the MACS is sometimes called "the potato-peeling problem."

Very few algorithms for finding MACS in digital images have been published. Often people have been content to find the maximal enclosed disc/ball. This can easily be found by computing the distance transform of the object. The maximal distance value is the centre of the maximal inscribed disc/ball [2,3]. An overview of the state of the MACS art can be found in [4], where only 2D is discussed. A discrete algorithm for finding the convex subset that minimizes the Hausdorff distance between the object and the subset can be found in [5]. It generates a reasonable approximation of MACS. In the case where the object is bounded by n-sided polygon (all discrete objects can be easily transformed to such a polygon) a $O(n^7)$ solution is found in [6]. It can only be used for very small objects. In [4] an approximation of this solution is given for star-shaped n-sided polygons. The approximation presented here is much coarser than those mentioned above,

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 438–445, 2005.

but it can be used for arbitrary connected objects in both 2D and 3D, i.e., the objects can have holes (2D) or cavities and tunnels (3D).

The CH can be used to identify concavities of an object by computing the convex deficiency, i.e. by subtracting the object from its convex hull. Similarly, the MACS can be used to identify protrusions. It can also be found to identify the main "body" of the object. A typical application is to find the body of a cell and ignore (or identify) any thin protrusions. In our case, the need for computing the MACS came up in two applications: finding the main body of pores in paper and the main body of proteins consisting of several protruding subparts [7].

An object with a large convex deficiency can be rather close to being convex, close in the sense that the MACS includes most of the object area (think of a convex object with a few single thin protrusions or some holes near the edges). Our MACS approximation is intended for objects in 2D or 3D having a large MACS. In these cases it will usually produce good approximations. If the object is, e.g., snakelike, the approximation can become far from the true MACS, but it is hard to think of applications where identifying the MACS would be valuable for such objects.

Our algorithm is based on a simple idea: use an approximation of the CH to identify concavities in the object and then iteratively remove them, starting with the deepest one. The depths of the concavities are computed using a distance transform of the convex deficiency from the background, constrained by the object. The CH approximation and constrained distance transform can be computed by simple local operations. The deepest concavity is removed by cutting the object into two parts using a straight line. The cutting is done so as to remove as little of the object as possible. This cutting is not local, but only simple operations are needed. The remaining part of the object is "more convex" than the original object. The process is repeated until no concavities remain. The remaining part of the object is the approximation of the MACS.

2 Notations

Since it will always be clear from the text if 2D or 3D images are considered, both pixels (2D) and voxels (3D) are denoted *elements*. Let I be a binary image containing the two sets B and W, where B is the set of black elements (the object) and W is the set of white elements (the background).

In 2D, 4-connectivity is used for W and 8-connectivity is used for B. Two elements are 4-connected if they share an edge and 8-connected if they share an edge or a vertex. In 3D, 6-connectivity is used for W and 26-connectivity for B. Two elements are 6-connected if they share a face and 26-connected if they share a face, an edge, or a vertex. Pairwise connected elements are referred to as *neighbours*. A set of elements is connected if, for any two elements a and b in the set, there is a path between a and b consisting of neighbouring elements all belonging to the set.

We assume that B is a connected set of elements. If this is not the case, each connected component must handled separately in the algorithm.

A white element sharing a side (2D) or a face (3D) with at least one black element is referred to as a *border element*. A border element will be defined as being in a local concavity by the configuration of its neighbouring black elements.

To find a globally deepest concavity, we compute the convex hull (CH). It is approximated by a *covering polygon* (2D) or a *covering polyhedron* (3D). The term *envelope* will be used to denote either covering polygon or covering polyhedron. The elements in the convex deficiency, i.e., the elements in the envelope that are not object (black) will be marked as gray, G.

The set of element(s) that are found to be the deepest concavity is denoted C. These corresponds to the reflex points in [4].

The end result of the algorithm is an approximation of the largest convex set enclosed in the object, i.e., the maximal area/volume convex subset (MACS).

3 Computing the Envelope

There are many algorithms for computing the CH of a digital object. We will use the one in [8], that approximates the CH by a covering polygon (polyhedron). It is simple and uses only local operations. The envelope is obtained by iteratively filling local concavities. The resulting shape is convex (coarse approximation) or very nearly convex (better approximation where some very shallow concavities may remain). Brief descriptions of the [8] algorithm in 2D and 3D are given in the subsections of this section.

The number of possible orientations of the sides (faces) of the envelope depends on the size of the neighbourhood from which curvature information is derived: the larger neighbourhood, the larger number of sides (faces) of the envelope. For example, in 2D, if curvature information is derived from the 3×3 neighbourhood of a border element, eight orientations are possible for the sides of the envelope, so what is actually computed is the minimal covering octagon. In 3D, using the $3 \times 3 \times 3$-neighbourhood will result in the smallest covering rhombicubeoctahedron (an Archimedian solid with 26 faces). If curvature information is gathered from a larger neighbourhood, many more orientations are possible (in fact, infinitely many, but still far from *all* orientations). The extension to larger neighbourhoods is achieved by a labeling procedure using 3×3-neighbourhoods ($3 \times 3 \times 3$-neighbourhoods).

After the computation of the envelope the image will be ternary: object elements (black), B, elements in the envelope that are not object elements, i.e. the convex deficiency, (gray), G, and background elements (white), W.

Computing the 2D Convex Hull. The 2D approximation of the convex hull is computed as follows. We give the two versions used in this paper.

3×3 Case

1. Iteratively change border elements with more than three non-white neighbours to gray.

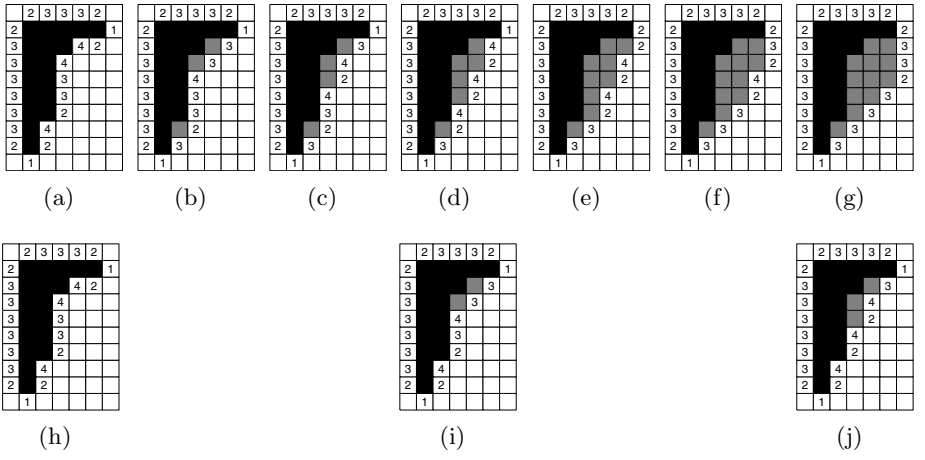


Fig. 1. Computing the covering polygon of an object using the 3×3-rule, (a)-(g); and using the 7×7-rule (h)-(j)

7×7 Case

1. Each border element is labeled with the number of its non-white neighbours.
2. Border elements labeled 3 with no neighbour labeled less than 3 are marked.
3. Change the following border elements to gray: elements labeled more than 4; elements labeled 4 with at least one neighbour labeled more than 3, or two neighbours labeled 3, or one marked neighbour.
4. Repeat from (1) until no changes occur.

A small example of how these two algorithms work is found in Figure 1.

Computing the 3D Convex Hull. The 3D approximation of the convex hull is computed as follows. We give the two versions used in this paper.

$3 \times 3 \times 3$ Case

1. For each border element, its non-white face and edge neighbours in the x-, y-, and z-planes are counted, respectively. If the maximum of these three sums is larger than 3, the element is marked.
2. Marked elements are changed to gray.
3. Repeat from (1) until no changes occur.

$5 \times 5 \times 5$ Case

1. For each border element, its non-white face and edge neighbours in the x-, y-, and z-planes are counted, respectively. These three sums, denoted Σ_x, Σ_y, and Σ_z, are stored as labels of the border element.

2. Change the following border elements to gray: Elements with at least one $\Sigma_k > 4$, ($k \in \{x, y, z\}$); elements with one $\Sigma_k = 4$ and having, in the same k-plane, at least one neighbour with $\Sigma_k > 2$.
3. Repeat from (1) until no changes occur.

4 Finding the Deepest Concavity

Here we describe how to find the places where the object must be cut to make it convex. To find the deepest concavity C, we use a constrained distance transform. In a distance transform, each element in one set is labeled with the distance to the closest element in another set. In a constrained distance transform [9], a third set acts as a barrier for the distance propagation. In our case, we will compute the distance transform of the convex deficiency, G, from the background, W, with the object, B, as the constraint.

We use *weighted* distance transforms. These are simple to compute and use, while being reasonably rotation independent distance, see, e.g., [2] for 2D and [3] for 3D. The distance between two elements is defined by the shortest path between them, using only steps between neighbours and weighting the steps according to the neighbouring relation. In 2D, we use the weights 3 and 4 for edge- and vertex-neighbours, respectively; and in 3D we use the weights 3, 4, and 5 for face-, edge- and vertex-neighbours, respectively. The actual computation is done by the efficient chamfer algorithm, see [2,3], again. The number of necessary scans depends on the object configuration.

The constrained distance transform of the gray pixels in the small example from Figure 1 is found in Figure 2. The deepest concavity C is simply the element with the largest distance label, in Figure 2(a) 8 and in Figure 2(b) 3.

Often neighbouring elements gets the same distance label, therefore connected components (using the background connectivity, as the gray elements are part of the original background) with the same distance label are considered together, so C consists of more than one element. If 6 were the maximum label in Figure 2(a), C would consist of the two elements labeled 6. If there is more than one connected component with elements having maximal distance label, as in Figure 2(b), anyone of them can be arbitrarily chosen as C.

(a) (b)

Fig. 2. Constrained distance transforms of the gray elements in Figure 1(g),(j): each gray element is labeled with the shortest distance to a white element

If there are holes (cavities) in the object they should be treated before the concavities. Each hole is treated as a connected set. Compute the weighted distance transform of the object *and* the holes from the non-black elements outside the object. The depth of each hole is the maximum distance value of any of the elements in the hole. The deepest hole (cavity) becomes C.

In 3D, there can also be tunnels. They do not need any special treatment, as they will become part of G when computing the envelope and will thus get distance values. In this case, C may not be border elements, as the elements farthest from the background may be in the interior of a tunnel. However, when this C is cut off, the tunnel will be broken and next time C will be a "normal" concavity.

5 Removing a Concavity

The (set of) deepest concavity element(s), C, has been found. It is removed by cutting the object into two (or more) parts, using straight lines (planes) through C. One part will be discarded and the other(s) will be kept for further processing. The aim is to remove as little of the object as possible, therefore cuts are made in all of a fixed number of directions and the one leaving the largest number of elements is chosen. In 2D we use 8 directions (angles $n\pi/4, n = 0, \ldots, 7$), and in 3D 18 directions (defined later). This is in agreement with envelopes based on the 3×3-rule ($3 \times 3 \times 3$-rule.)

In 2D, the line that separates the object and C must satisfy one of the following equations $y = m$, $y = x + m$, $x = m$, or $y = -x + m$. The values of m are computed such that one connected component of the object is on one side of the line and C is on the other. In Figure 2(a), C is the element with distance label 8. In Figure 3 all eight possible cuts removing C are shown.

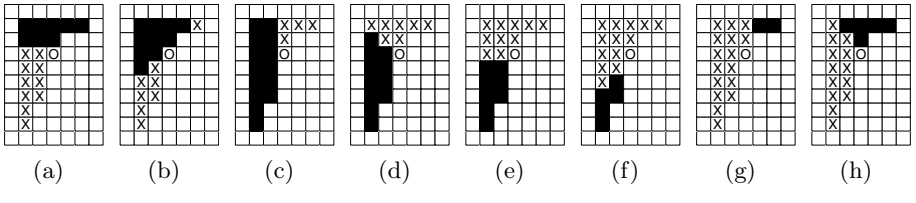


Fig. 3. The cuts induced by the eight directions applied to the running example. The element marked "O" is the deepest concavity. The elements removed by each cut are marked "X", while black elements are the ones remaining.

In some cases, the cut splits the object into two connected components. In this case, the size of part on the "C-side" measures how much is removed. In other cases, where the object "protrudes" on both sides of C the line splits the object into three (or more) connected components, see Figure 3(b). In these cases, only the smallest of the components on the C-side is removed and its size is the measure of the cut. The protrusion on the other side of C will probably

be removed at a later stage, but not necessarily *all* of it, so removing only one component can increase the size of the final MACS.

After cuts have been made in all eight directions, the cut removing the smallest connected component is chosen to remove C. In Figure 3 this is the cut in (b); only 1 element is eliminated.

In 3D, the cutting plane must satisfy one of the nine equations $x = m$, $y = m$, $z = m$, $y = x+m$, $y = -x+m$, $z = x+m$, $z = -x+m$, $z = y+m$, or $z = -y+m$. The number of different planes that must be tested becomes 18. The cutting is done exactly as in the 2D case – for each direction m is determined and the smallest connected component after the cut is considered for removal. In the end the best of the 18 possible cuts is chosen to remove C.

6 Finding the Maximal Convex Subset

By sequentially repeating the algorithms described in Section 3-5, MACS is computed. In each iteration, first the envelope is computed using the 3×3-rule ($3 \times 3 \times 3$-rule) and the deepest concavity C is found. If the distance value of $C > 3$, C is removed by making the best cut.

If, however, the distance value of $C = 3$, the object is possibly already convex. Remember that the envelopes using the $3 \times 3(\times 3)$ neighbourhoods overestimate the true CH, because of the limited number of directions of edges (faces). Thus, if $C = 3$ we recompute the envelope, using the more accurate 7×7-rule ($5 \times 5 \times 5$-rule). If any concavities do remain, using these rules, they must be removed in the usual way.

Note that, the envelope should not be computed using the larger neighbourhoods. Experimentation has shown that, in most cases, using more directions when computing the envelope than when cutting will remove more than is necessary of the object. However, using the larger neighbourhoods at the very end can "save" many object elements in the MACS, as each "pseudo-concavity" can lead to the cutting off of many elements.

7 Examples and Conclusion

In Figure 4 there are a number of examples of how the algorithm works, both synthetic and real. In Figure 4(a) we have small fish with an eye-hole and in (b) its MACS. We mentioned cells with thin extensions as suitable for the algorithm. Figure 4(c) shows a nerve cell with axons from a stem cell project at our lab. The cell body is found by computing the MACS, Figure 4(d). Figure 4(e) shows a synthetic 3D object, a "potplant" and (f) its MACS. Not all of the pot remains, as the number of cutting plane directions is limited, but most of it is there. Figure 4(g) shows a pore (void) segmented from a 3D image of a piece of milk cartoon. The pore shape is very irregular with deep concavities, more than is apparent from the image. Finding the body of such pores was one of the motivations of this work. The MACS of the pore is shown in Figure 4(h).

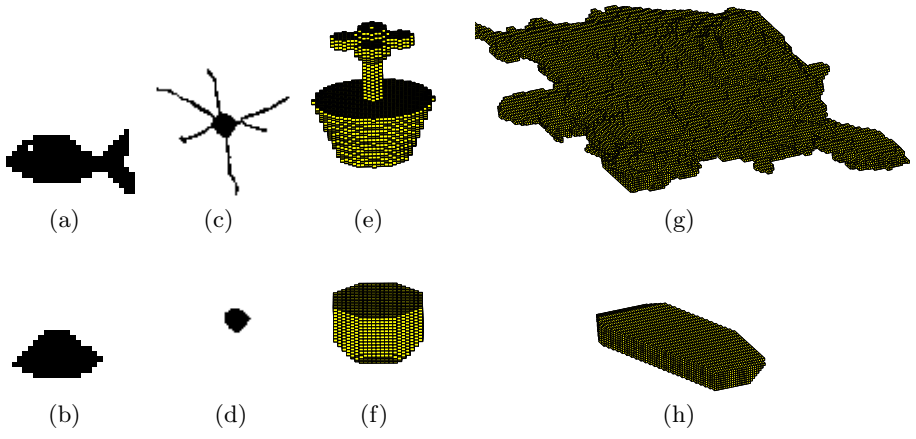


Fig. 4. Examples of maximal inscribed convex sets in 2D and 3D. See text

Even if the proposed algorithm is not perfect, it is shown to be a useful tool for finding an approximation of the largest inscribed convex set. It is simple to implement and uses mostly local operations, so it is reasonably fast even in 3D.

References

1. Preparata, F.P., Shamos, M.I.: Computational Geometry an Introduction. Springer-Verlag, New York (1985)
2. Borgefors, G.: Distance transformations in digital images. Computer Vision, Graphics, and Image Processing **34** (1986) 344–371
3. Borgefors, G.: On digital distance transforms in three dimensions. Computer Vision and Image Understanding **64** (1996) 368–376
4. Chassery, J.M., Coeurjolly, D.: Optimal shape and inclusion. In Ronse, C., Najman, L., Decencière, E., eds.: Mathematical Morphology: 40 Years On. Computational Imaging and Vision, Springer, Dordrecht (2005) 229–248
5. Chassery, J.M.: Discrete and computational geometry approaches applied to a problem of figure approximation. In Pietikäinen, M., Röning, J., eds.: Proc. 6th Scandinavian Conference on Image Analysis, Oulo, Finland (1989) 856–859
6. Chang, J.S., Yap, C.K.: A polynomial solution for the potato-peeling problem. Discrete & Computational Geometry **1** (1986) 155–182
7. Sintorn, I.M.: Segmentations methods and shape descriptions in digital images – applications in 2D and 3D microscopy. PhD thesis, Swedish University of Agricultural Sciences (2005)
8. Borgefors, G., Sanniti di Baja, G.: Analyzing nonconvex 2D and 3D patterns. Computer Vision and Image Understanding **63** (1996) 145–157
9. Piper, J., Granum, E.: Computing distance transformations in convex and non-convex domains. Pattern Recognition **20** (1987) 599–615

Computing Homographies from Three Lines or Points in an Image Pair*

G. López-Nicolás, J.J. Guerrero, O.A. Pellejero, and C. Sagüés

DIIS - I3A, Universidad de Zaragoza.
C/María de Luna 1, E-50018 Zaragoza, Spain
{gonlopez, jguerrer, csagues}@unizar.es

Abstract. This paper deals with the computation of homographies from two views in a multi-plane scene. In the general case, homographies can be determined using four matched points or lines belonging to planes. We propose an alternative method when a first homography has been obtained, and then three matches are sufficient to compute a second homography. This process is based on the geometric constraint introduced by the first homography. In this work, the extraction and matching of features, points or lines, is automatically performed using robust techniques. Experimental results with synthetic data and real images show the advantages of this approach. Besides, the performance using points or lines as image features is compared.

Keywords: Homographies, multi-plane scenes, multi-view constraints, point and line matching.

1 Introduction

The algorithm we are going to describe deals with scenes containing planar surfaces. These are characteristic of urban scenes and other man made environments, in which lines and planes are plentiful elements. In these cases, the recovery of the 3D geometry can be supported by homographies computation.

Several authors detect and use planar homographies in image pairs [1], [2], but the detection of different homographies is considered as independent processes. There, the estimation of each homography is carried out independently of the others repeating the same algorithm each time.

The knowledge of one homography in an image pair impose restrictions on the other homographies. So the collection of homographies of multiple planes between two views spans a 4-dimensional linear subspace [3]. However, this constraint requires the number of planes in the scene to be greater than four. Other works [4], [5] apply the constraint replacing the need for multiple planes by the need for multiple views. They only need two planes, and often a single one is enough under restricted assumptions on camera motion, but more than four views are necessary.

* This work was supported by project DPI2003-07986.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 446–453, 2005.

In previous works [6], we address the estimation of two or more homographies in an image pair and the computation of the fundamental matrix through them, using straight lines extracted and matched as described in [7]. It has been reported that the multi-plane algorithm is not as stable as the general method to compute the fundamental matrix [1], but when less than three planes are observed, which is quite usual in man made environments, the multi-plane algorithm gives similar and even better results than the general method.

This work is focused in the case of two views. Usually, more than four planes are not available in the scene, so it is necessary to use different constraints from the ones presented in [5]. Here we present the computation of the second homography using the constraints imposed by a first one, through the homology of an image into itself. A complete algorithm which uses these constraints, matching lines or points automatically, is presented. Experimental results with synthetic data and real images show the advantages of this approach.

2 Constraints on Homographies

The computation of a homography from three point correspondences in an image pair knowing the fundamental matrix is presented in [8]. Now we show that a similar process can also be carried out, but using another homography instead of the fundamental matrix.

In order to compute a first homography $\mathbf{H}_{21}^{\pi_1}$ for a plane π_1 we need at least four matched points or lines belonging to that plane. Once this homography is obtained, a second homography $\mathbf{H}_{21}^{\pi_2}$ for a plane π_2 can be determined with only three points or lines using the constraints introduced by the first homography.

A useful method for real applications needs to be robust. We have chosen the RANSAC method, which is robust in presence of outliers. The presented approach reduces the sample size needed to have a minimum set of data required to instantiate the free parameters of the model. The number of samples necessary to ensure that at least one of them has no outliers depends on the sample size. Therefore, a shorter sample size leads us to a faster algorithm.

2.1 Working with Points

The method for computing a second homography from three points consists in obtaining a fourth point using the constraints provided by the first homography and the idea of plane parallax [8]. Once we get the set of four points, the general algorithm can be used to obtain the second homography.

Let us suppose the computed homography ($\mathbf{H}_{21}^{\pi_1}$) induced by plane π_1 and the projections in both images ($\mathbf{p}_1, \mathbf{p}_2$) of a point ($\mathbf{p}$) belonging to the second plane π_2 (Fig. 1a). The corresponding of $\mathbf{p}_1$ through the homography $\mathbf{H}_{21}^{\pi_1}$ is $\bar{\mathbf{p}}_2 = \mathbf{H}_{21}\mathbf{p}_1$. The epipolar line joining $\mathbf{p}_2$ to $\bar{\mathbf{p}}_2$ is the epipolar line of $\mathbf{p}$ in the second image. Repeating the process with another point $\mathbf{q}$, the line joining $\mathbf{q}_2$ to $\bar{\mathbf{q}}_2 = \mathbf{H}_{21}^{\pi_1}\mathbf{q}_1$ is obtained. The intersection of these two lines determine the epipole $\mathbf{e}_2$. This is the fourth point used for the computation of the second homography.

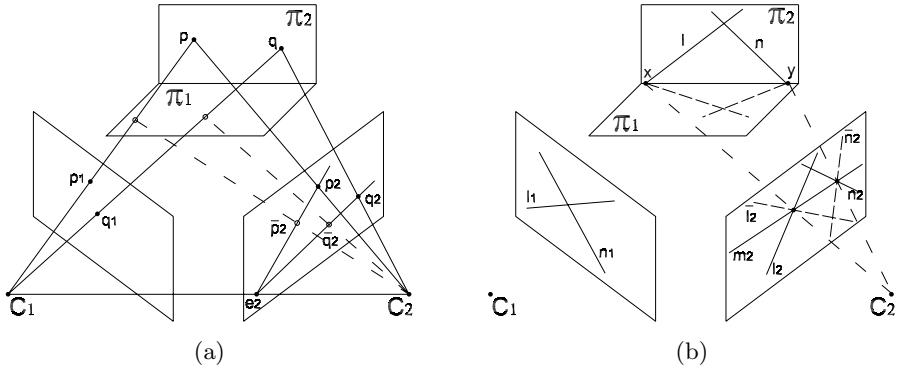


Fig. 1. Geometric method to obtain a second homography using points (a) or lines (b)

Two matched points are sufficient to get the epipole. As three points are available, three epipolar lines can be obtained. The intersection of these lines is the epipole, but they will not intersect exactly in a point due to image noise. So, the epipole is computed as the centroid of the triangle defined by these three lines.

A homography cannot be computed if three of the four correspondences are collinear. Therefore, the method for determining the second homography fails if one of the three points is the epipole or if two of the points are collinear with the computed epipole. Besides, when points are nearly collinear with the epipole, the method will give a poorly conditioned estimation of the homography.

2.2 Working with Lines

The process is similar using straight lines instead of points. Three lines are available, if we find the fourth line, a second homography will be determined.

Line $\mathbf{l}$ belonging to the plane π_2 is projected in both views as $(\mathbf{l}_1, \mathbf{l}_2)$ (Fig. 1b). From the homography $\mathbf{H}_{21}^{\pi_1}$ induced by π_1, we can obtain the corresponding of this line in the second image as $\bar{\mathbf{l}}_2 = (\mathbf{H}_{21}^{\pi_1})^{-T}\mathbf{l}_1$. The intersection of lines $\mathbf{l}_2$ and $\bar{\mathbf{l}}_2$ is the projection of the point $\mathbf{x}$ in the second image. In the same way, the process is repeated with another line, $\mathbf{n}$. From the intersection of lines $\mathbf{n}_2$ and $\bar{\mathbf{n}}_2 = (\mathbf{H}_{21}^{\pi_1})^{-T}\mathbf{n}_1$ the projection of the point $\mathbf{y}$ in the second image is obtained.

Points $\mathbf{x}$ and $\mathbf{y}$ belong to both planes π_1 and π_2. The line passing through the projection of these points $\mathbf{m}_2$, is the projection of the intersection of the planes. Therefore, as this line belongs to π_2, it can be used as the fourth line to compute the second homography.

Summarizing, each line gives a point belonging to the intersection of the planes. With two lines the intersection line is determined. In our case, three lines are available and therefore, three points define the intersection line. As the line will not pass exactly through these points due to image noise a least squares solution is used for computing this line.

This method presents degeneracies if one of the other three lines is the intersection line of both planes, or if the obtained line is parallel or intersects in the same point with the two other lines.

Working with lines give us a stronger constraint than using points when we have two homographies and we want to compute the third. The method can be applied twice obtaining two intersection lines, and it only needs two initial matched lines to compute the third homography. If we work with points, we need three points again, because the epipole is unique for the image pair. This difference between points and lines is because less than three points do not define a spatial plane, while two lines are enough.

3 Computing Through the Homology

In the previous section, we have described a geometric method to compute a second homography $\mathbf{H}_{21}^{\pi_2}$ when a first homography $\mathbf{H}_{21}^{\pi_1}$ is known and three matched points or lines belonging to a second plane π_2 are available. Next, another alternative method to compute a second homography is presented. The main idea is the use of the planar homology, which is a well-known model from two planes projected in two views.

A planar homology is a plane projective transformation with five degrees of freedom. It has a line of fixed points, called the axis, and a fixed point not on the line, called the vertex (Fig. 2). Planar homologies arise naturally in an image when two planes related by a perspectivity in 3-space are imaged [9].

The projective transformation representing the homology can be parameterized directly [10] in terms of the vector representing the axis $\mathbf{a}$, the vector representing the vertex $\mathbf{v}$, and the characteristic cross-ratio μ as $\mathbf{H} = \mathbf{I} + (\mu - 1)\frac{\mathbf{v}\,\mathbf{a}^T}{\mathbf{v}^T\mathbf{a}}$, being $\mathbf{I}$ the identity matrix. It can also be parameterized [5] as $\mathbf{H} = \mathbf{I} + \mathbf{v}\,\mathbf{a}^T$, which implies $\mu = \mathbf{v}^T\mathbf{a} + 1$. The cross ratio μ is formed by corresponding points, the vertex and the intersection of their join with the axis, and it is an invariant of the homology.

A homology has two unary eigenvalues, the third eigenvalue coincide with the cross ratio μ. Eigenvectors of the homology associated to double unary eigenvalue span the axis $\mathbf{a}$, and the other eigenvector, associated to eigenvalue μ, give us the vertex $\mathbf{v}$ [5].

From two homographies a homology can be computed as $\mathbf{H} = \mathbf{H}_{21}^{\pi_2}(\mathbf{H}_{21}^{\pi_1})^{-1}$. First homography $\mathbf{H}_{21}^{\pi_1}$ has already been obtained and we want to compute second homography $\mathbf{H}_{21}^{\pi_2}$. The main idea in this second method is to determine a homology using the available features. A homology can be computed from three matched features since it has five degrees of freedom and each feature gives two constraints.

We can map the point $\mathbf{p}_1$ (Fig. 2a) in the second image through the first homography $\mathbf{H}_{21}^{\pi_1}$ using $\bar{\mathbf{p}}_2 = \mathbf{H}_{21}^{\pi_1}\mathbf{p}_1$. So, we obtain three matched points $(\mathbf{p}_2, \bar{\mathbf{p}}_2)$, $(\mathbf{q}_2, \bar{\mathbf{q}}_2)$ and $(\mathbf{r}_2, \bar{\mathbf{r}}_2)$. Since the homology is $\mathbf{H} = \mathbf{H}_{21}^{\pi_2}(\mathbf{H}_{21}^{\pi_1})^{-1}$, we have for the first match $\mathbf{p}_2 = \mathbf{H}\bar{\mathbf{p}}_2$ and so forth. From the three matches, the vertex $\mathbf{v}$ and the axis $\mathbf{a}$ are obtained geometrically, the cross ratio is computed too, and from

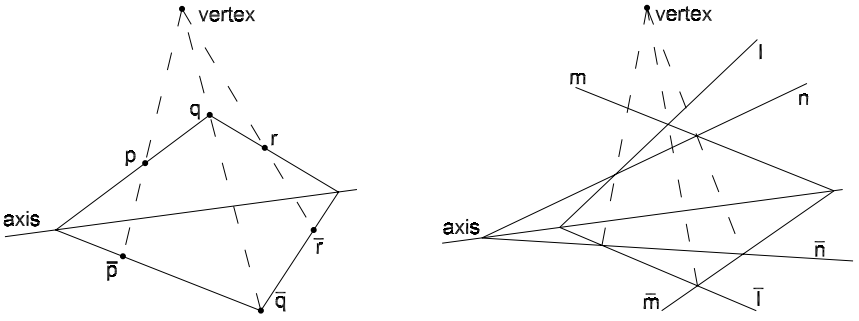


Fig. 2. Correspondences to compute a planar homology from points (a) or lines (b)

them the homology is computed using the previous parameterization. Finally, we can work out the value of the second homography from $\mathbf{H}_{21}^{\pi_2} = \mathbf{H}\mathbf{H}_{21}^{\pi_1}$.

The process is similar using lines (Fig. 2b). The mapping of line $\mathbf{l}_1$ in the second image through $\mathbf{H}_{21}^{\pi_1}$ is carried out using equation $\bar{\mathbf{l}}_2 = (\mathbf{H}_{21}^{\pi_1})^{-T}\mathbf{l}_1$. From here, three matches are obtained: $(\mathbf{l}_2, \bar{\mathbf{l}}_2)$, $(\mathbf{m}_2, \bar{\mathbf{m}}_2)$ and $(\mathbf{n}_2, \bar{\mathbf{n}}_2)$. Finally, second homography is computed as in the previous case.

4 Experimental Results

The proposed approach has been experimentally validated using synthetic data and real images. In all the experiments, the first homography has been automatically obtained and subsequently fixed, in order to avoid its variability (due to RANSAC) to affect the behavior of the computation of the second homography, which is analyzed.

The second homography is computed using all the methods presented: The general method of four features, the three features method and through the homology. The experiments are carried out with points and lines.

With real images the matches are automatically obtained for points and lines. In the case of points, they are extracted by Harris corner detector and matched by correlation and relaxation algorithms [11]. The process of extract and match lines, which is based in geometric and brightness parameters, is explained in [7]. The images of a house and a college with the matched features are shown in Fig. 3. The synthetic scene consists of random points in one case and lines on the other. These virtual features have white noise of $\sigma = 0.3$ pixels and are distributed in three perpendicular planes.

Several criteria can be used to measure the accuracy of the obtained homography. We measure the first order geometric error computed as the Sampson distance [8] for a set of 20 matched points manually extracted and matched. Each experiment is repeated 100 times and the mean and median error is shown in Table 1. The accuracy obtained with all the methods is similar, demonstrating the validity of the proposed approach. The mean is always a little higher than

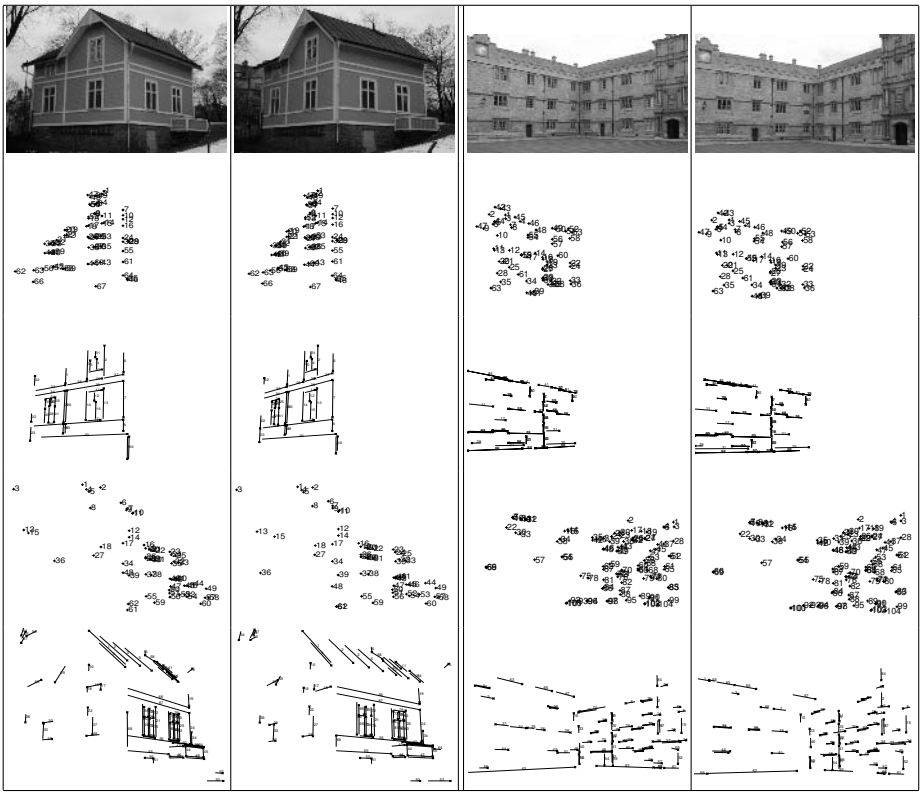
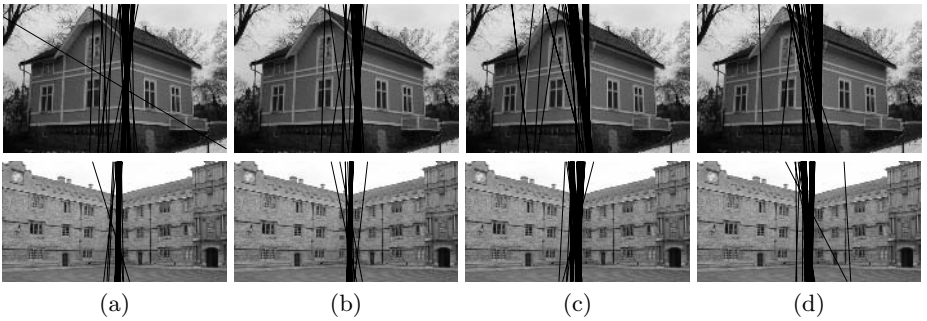


Fig. 3. Images of the house and college (1st row). Matches corresponding to the first homography with points (2nd row) and lines (3rd row). Putative matches available to search the second homography with points (4th row) and lines (5th row). Original images from KTH (Stockholm) and VGG (Oxford)



(a) (b) (c) (d)

Fig. 4. Intersection of the planes through the eigenvalues of the homology. Lines corresponding to 100 executions are represented using 4 points (a); 3 points (b); 4 lines (c) and 3 lines (d)

Table 1. Sampson distance of the second homography for 20 points manually matched. We show in 100 repetitions the mean and median error and the mean execution time. Results are obtained using four points (4p) or lines (4l); three points (3p) or lines (3l); and through homology with three points (3p+H) or lines (3l+H)

		4p	3p	3p+H	4l	3l	3l+H
Synthetic data	Mean error (pixels)	0.538	0.554	0.531	0.537	0.515	0.427
	Median error (pixels)	0.542	0.559	0.532	0.541	0.526	0.407
	Mean time (seconds)	6.94	1.94	1.66	4.21	1.50	1.43
Images of the house	Mean error (pixels)	1.230	1.191	1.206	1.257	1.591	1.576
	Median error (pixels)	1.168	1.171	1.162	1.134	1.099	0.821
	Mean time (seconds)	3.50	1.48	1.29	4.24	1.74	1.58
Images of the college	Mean error (pixels)	0.981	0.942	0.933	0.747	0.737	0.718
	Median error (pixels)	0.854	0.755	0.782	0.789	0.724	0.740
	Mean time (seconds)	4.08	1.43	1.48	4.71	0.92	1.15

the median, this is because the presence of some spurious iterations. Sampson distance obtained with the four feature method is similar to the three features method, but it needs more iterations to get this similar result. Both methods of three features and using the homology can be used either because the obtained results are nearly equal.

There is almost no differences between errors obtained using points or lines. Although it depends on the extracted and matched features. In our experiments, the initial number of matches is similar for points and lines, and they are homogeneously distributed over the images. The use of lines instead of points has some advantages, mainly in man made environments. Straight lines can be accurately extracted in noisy images, they capture more information than points and they may be matched where partial occlusions occur.

Table 1 also shows the mean execution time of the iterations. One of the advantages of the three features method is that it consumes less time than using four features. This is because the number of samples necessary to avoid outliers with a high probability is exponential in the size of the sample set.

The epipole and the intersection of the planes can also be computed from the homology. The epipole is the eigenvector corresponding to the non-unary eigenvalue and the other two eigenvectors define the intersection line of the planes. In Fig. 4 the intersection line of the planes is shown for 100 repetitions. Only in a small percent of cases it is wrong, which is coherent with the probability of fail assumed in the robust method. Besides, in the same way that the features on the first plane have been automatically segmented (second and third row of Fig. 3), the method gives a robust selection of the features on second plane [11].

5 Conclusions

We have presented a method to compute a second homography from three points or lines in an image pair. In other works, each homography is computed indepen-

dently of the others, and four features are needed. But once a first homography is obtained we can take advantage of the constraints imposed to determine another homography.

Experiments have been carried out in an automatic way using points and lines as features. Although the use of points is more common, lines has proven to give similar results, having advantages in man made environments.

The main advantage of this approach is the computing time reduction for the second homography. This is due to the use of a smaller size of the sample, which is three instead of four. This is useful to improve the performance of real time applications. Besides, the accuracy of the obtained homographies with the three features method is as good as using four.

References

1. Luong, Q., O.D.Faugeras: Determining the fundamental matrix with planes: Unstability and new algorithms. In: In Proc. Conference on Computer Vision and Pattern Recognition. (1993) 489–494
2. Vincent, E., Laganière, R.: Detecting planar homographies in an image pair. In: Proc. Second International Symposium on Image and Signal Processing and Analysis. (2001) 182–187
3. Shashua, A., Avidan, S.: The rank 4 constraint in multiple (≥ 3) view geometry. In: Proc. European Conf. Computer Vision, St. Louis, Missouri (1996) 196–206
4. Zelnik-Manor, L., Irani, M.: Multiview subspace constraints on homographies. In: Proc. Int. Conf. Computer Vision, St. Louis, Missouri (1999) 710–715
5. Zelnik-Manor, L., Irani, M.: Multiview constraints on homographies. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 214–223
6. Pellejero, O., Sagüés, C., Guerrero, J.: Automatic computation of fundamental matrix from matched lines. In: Current Topics in Artifical Intelligence, LNCS-LNAI 3040. (2004) 197–206
7. Guerrero, J., Sagüés, C.: Robust line matching and estimate of homographies simultaneously. In: IbPRIA, Pattern Recognition and Image Analysis, LNCS 2652. (2003) 297–307
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
9. Van-Gool, L., Proesmans, M., Zisserman, A.: Grouping and invariants using planar homologies. In: Workshop on Geometrical Modeling and Invariants for Computer Vision. (1995)
10. Criminisi, A., Reid, I., Zisserman, A.: Duality, rigidity and planar parallax. In: European Conference on Computer Vision, LNCS 1407. (1998) 846–861
11. López-Nicolás, G., Sagüés, C., Guerrero, J.: Automatic matching and motion estimation from two views of a multiplane scene. In: IbPRIA, Pattern Recognition and Image Analysis. (2005) To appear, LNCS.

Commute Times, Discrete Green's Functions and Graph Matching

Huaijun Qiu and Edwin R. Hancock

Department of Computer Science,
University of York, York, YO10 5DD, UK

Abstract. This paper describes a graph-spectral method for simplifying the structure of a graph. Our starting point is the lazy random walk on the graph, which is determined by the heat-kernel of the graph and can be computed from the spectrum of the graph Laplacian. We characterise the random walk using the commute time between nodes, and show how this quantity may be computed from the Laplacian spectrum using the discrete Green's function. Our idea is to augment the graph with an auxiliary node which acts as a heat source. We use the pattern of commute times from this node to decompose the graph into a sequence of layers. These layers can be located using the Green's function. We exploit this decomposition to develop a layer-by-layer graph-matching strategy. The matching method uses the commute time from the auxiliary node as a node-attribute.

1 Introduction

Spectral graph theory [2] is concerned with characterising the structural properties of graphs using information conveyed by the eigenvalues and eigenvectors of the Laplacian matrix (the degree matrix minus the adjacency matrix). One of the most important tasks that arises in the analysis of graphs is that of how information flows with time across the edges connecting nodes. This process can be characterised using the heat equation [5]. The solution of the heat equation, or heat kernel, can be found by exponentiating the Laplacian eigensystem over time. The heat kernel contains a considerable amount of information concerning the distribution of paths on the graph. For instance, it can be used compute the lazy random walk on the nodes of the graph. It may also be used to determine commute times under the random walk between pairs of nodes. An alternative, but closely related, characterisation of the graph is the discrete Green's function which captures the distribution of sources in the heat flow process. Not surprisingly, there is a direct link between commute times and the Green's function [3].

Random walks [12] have found widespread use in information retrieval and structural pattern analysis. For instance, the random walk is the basis of the Page-Rank algorithm which is used by the Googlebot search engine [1]. In computer vision random walks have been used for image segmentation [8] and clustering [11]. More recently both Gori, Maggini and Sarti [4], and, Robles-Kelly and Hancock [10] have used random walks to sort the nodes of graphs for the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 454–462, 2005.

purposes of graph-matching. However, most of these methods use a simple approximate characterisation of the random walk based either on the leading eigenvector of the transition probability matrix, or equivalently the Fiedler vector of the Laplacian matrix [6]. However, a single eigenvector can not be used to determine more detailed information concerning the random walk such as the distribution of commute times.

The aim in this paper is to draw on more detailed information contained within the Laplacian spectrum, and to use the commute time as means of characterising graphs. Our overall aim is to develop a means of graph matching. Rather than using the string-like characterisations that result from the approximate random walks used by Gori, Maggini and Sarti, and, Robles-Kelly and Hancock we aim to develop one based on the concentric layers that result by repeatedly peeling away the boundary of the graph. The reason for this is that the pattern of concentric layers is less likely to be disturbed by structural noise than the random walk, which can be diverted. To address this problem using the apparatus of the heat equation, we augment the graph with an auxiliary node. This node is connected to each of the boundary nodes by an edge, and acts as a heat source. Concentric layers are characterised using the commute time from the auxiliary node. We show how to compute the commute times using the Green's function for the graph, and this may be effected using the Laplacian spectrum. We match graphs by separately matching the concentric layers.

2 Heat Kernel and Path-Weighted Matrix

Let the weighted graph Γ be the quadruple (V, E, Ω, ω), where V is the set of nodes, E is the set of arcs, $\Omega = \{W_u, \forall u \in V\}$ is a set of weights associated with the nodes and $\omega = \{w_{u,v}, \forall (u,v) \in E\}$ is a set of weights associated with the edges. Further let $T = diag(d_v; v \in V(\Gamma))$ be the diagonal weighted degree matrix with $T_u = \sum_{v=1}^{n} w_{u,v}$. The un-normalised weighted Laplacian matrix is given by $L = T - A$ and the normalized weighted Laplacian matrix is defined to be $\mathcal{L} = T^{-1/2}LT^{-1/2}$, and has elements $\mathcal{L}_{uv}(\Gamma) = \begin{cases} 1 & \text{if } u = v \\ -\frac{w_{u,v}}{\sqrt{d_u d_v}} & \text{if } u \neq v \text{ and } (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$.

The spectral decomposition of the normalised Laplacian is $\mathcal{L} = \Phi\Lambda\Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the diagonal matrix with the ordered eigenvalues as elements satisfying: $0 = \lambda_1 \leq \lambda_2 \ldots \leq \lambda_{|V|}$ and $\Phi = (\phi_1|\phi_2|....|\phi_{|V|})$ is the matrix with the ordered eigenvectors as columns.

In the paper we are interested in the heat equation associated with the graph Laplacian, i.e. $\frac{\partial \mathcal{H}_t}{\partial t} = -\mathcal{L}\mathcal{H}_t$ where $\mathcal{H}_t$ is the heat kernel and t is time. The solution of the heat-equation is found by exponentiating the Laplacian eigenspectrum i.e.$\mathcal{H}_t = \exp[-t\mathcal{L}] = \Phi \exp[-t\Lambda]\Phi^T$. The heat kernel is a $|V| \times |V|$ matrix, and for the nodes u and v of the graph Γ the element of the matrix is $\mathcal{H}_t(u,v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i(u)\phi_i(v)$.

Let us consider the matrix $P = I - \mathcal{L}$, where I is the identity matrix, then the heat kernel can be rewritten as $\mathcal{H}_t = e^{-t(I-P)}$. We can perform a McLaurin

expansion on the heat-kernel to re-express it as a polynomial in t. The result of this expansion is

$$\mathcal{H}_t = e^{-t(I-P)} = e^{-t}\left(I + tP + \frac{(tP)^2}{2!} + \frac{(tP)^3}{3!} + \cdots\right) = e^{-t}\sum_{r=1}^{\infty} P^r \frac{t^r}{r!} \quad (1)$$

Using the spectral decomposition of the normalised Laplacian, we have $P^r = (I-\mathcal{L})^r = \Phi(I-\Lambda)^r\Phi^T$ and as a result

$$P^r(u,v) = \sum_{i=1}^{|V|}(1-\lambda_i)^r\phi_i(u)\phi_i(v) = \sum_{\pi_r}\prod_i \frac{w(u_i,u_{i+1})}{\sqrt{d_{u_i}d_{u_{i+1}}}} \quad (2)$$

Hence, P^r can be interpreted as the sum of weights of all walks of length r joining nodes u and v. A walk π_r is a sequence of vertices $u_0,\cdots,u_r$ such that $u_i = u_{i+1}$ or $(u_i,u_{i+1}) \in E$.

3 Graph Derivation and the Multilayer Representation

We commence by constructing an augmented graph from the original graph by adding an auxiliary external node. We refer to this new graph as the *affixation graph*. It is constructed by connecting the additional node to each of the nodes on the boundary (or perimeter) of the original graph. Our aim in constructing this affixation graph is to simulate heat flow from the external node, which acts like an external heat source. We assign the label τ to the auxiliary node, and the *affixation graph* $\mathcal{A}(V^{'},E^{'})$ can be defined by $V^{'} = V \cup \{\tau\}$ and $E^{'} = E \cup \{(\tau,u), \forall u \in Boundary(\Gamma)\}$.

By analysing the heat-flow from the auxiliary node on the affixation graph, we can generate a multilayer representation of the original graph. The idea is to characterise the structure of the graph using the pattern of heat-flow from the source node. To embark on this study, let us first consider the relationship between the heat kernel and the lazy random walk.

Theorem 1. *The heat kernel is the continuous time limit of the lazy random walk.*

Proof. Consider a lazy random walk $R = (1-\alpha)I + \frac{W}{T}\alpha$ which migrates between different nodes with probability α and remains static at a node with probability $1-\alpha$, where W is the weighted adjacency matrix and T is the degree matrix.

Let $\alpha = \alpha_0\Delta t$ where $\Delta t = \frac{1}{N}$. Consider the distribution $R(V_N|V_0)$, which is the probability of the random walk joining node 0 and N, in the limit $\Delta t \to 0$

$$\lim_{N\to\infty} R^N = \lim_{N\to\infty}\left(I + (\frac{W}{T} - I)\alpha_0\frac{1}{N}\right)^N = e^{(\frac{W}{T}-I)\alpha_0} \quad (3)$$

while

$$\frac{W}{T} - I = T^{-1}A - I = T^{-1}(T-L) - I = -T^{-1}L \quad (4)$$

Now consider the discrete Laplace operator Δ with the properties: $\mathcal{L} = T^{1/2}\Delta T^{-1/2} = T^{-1/2}LT^{-1/2}$ *which implies:*

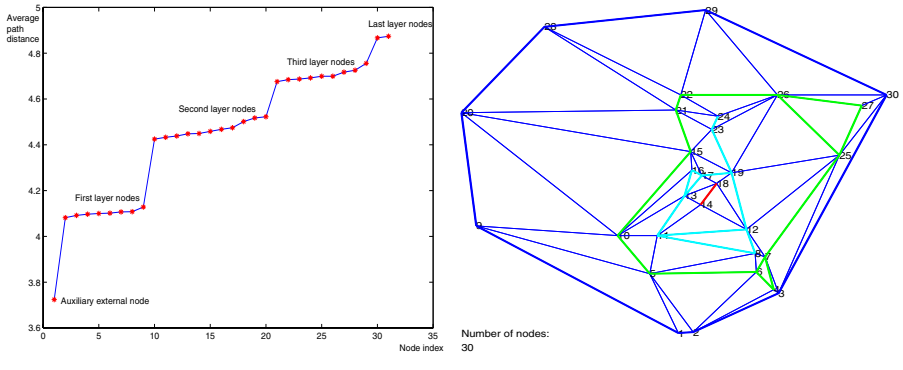
$$\Delta = T^{-1}L \tag{5}$$

Substituting Equation (4) and (5) into Equation (3), we get $\lim_{N\to\infty} R^N = e^{-\Delta\alpha_0}$ *which is just the expression for the heat kernel.* □

Corollary 1. *The path-weighted matrix $P^r(u,v)$ defined in Equation 2 is the sum of the probabilities of all the lazy random walks of length r connecting node u and v.*

Since P^r is the probability of the lazy random walk with a certain path length, we can make an estimate of the heat flow on the graph by taking the average value of P^r according to the path length r: $\mathcal{D}(u,v) = \frac{\sum_r rP^r(u,v)}{\sum_r P^r(u,v)}$. We take the external node τ to be the heat source and consider all the random walks starting from this point. The average path distance $\mathcal{D}(\tau,v)$ for all v in $V(\Gamma)$ follows a staircase distribution, which we can use to classify nodes into different layers.

Figure 1(a) illustrates this staircase property. The nodes with the same average distance correspond to the same layer of the graph. The corresponding multilayer graph representation is shown in Figure 1(b), where the nodes connected by edges of the same color belong to the same layer. Our matching process is based on the layers extracted in this way. To do this, we match the nodes in each layer in one graph to the nodes of the corresponding layer in a second graph. To do this we need a score-function to distinguish the different nodes in the same layer. Unfortunately, the average path distance can not be used for this purpose, since it is too coarsely quantised and can not be used to differentiate between the nodes in the same layer of a graph. We seek a



(a) Staircase distribution of the average path distance.

(b) An example of a multilayer graph.

Fig. 1. The staircase distribution and a multilayer graph

score function which is related to the heat kernel, and hence the heat-flow from the external source node, but offers us more distinct values for each individual node.

4 Score Function

Our score function is derived from the properties of the random walk. To commence, we note that the *hitting time* $Q(u,v)$ of a random walk on a graph is defined as the expected number of steps before node v is visited, commencing from node u. The *commute time* $O(u,v)$, on the other hand, is the expected time for the random walk to travel from node u to reach node v and then return. As a result $O(u,v) = Q(u,v) + Q(v,u)$. Our score function is the commute time between the external source node and the nodes of the original graph.

As we will demonstrate later, the commute time has some useful properties that render it suitable to our needs. First, we consider how the commute time can be computed.

The hitting time $Q(u,v)$ is given by [3]

$$Q(u,v) = \frac{vol}{d_v}G(v,v) - \frac{vol}{d_u}G(u,v)$$

where $vol = \sum_{v \in V(\Gamma)} d_v$ is the volume of the graph and function G is the Green's function on the graph.

The Green's function is the left inverse operator of the Laplace operator Δ, defined by $G\Delta(u,v) = I(u,v) - \frac{d_v}{vol}$. A physical interpretation of the Green's function is the temperature at a node in the graph due to a unit heat source applied to the external node. It is related with the heat kernel $\mathcal{H}_t$ in the following manner

$$G(u,v) = \int_0^\infty d_u^{1/2} \left(\mathcal{H}_t(u,v) - \phi_1(u)\phi_1(v)\right) d_v^{-1/2} dt$$

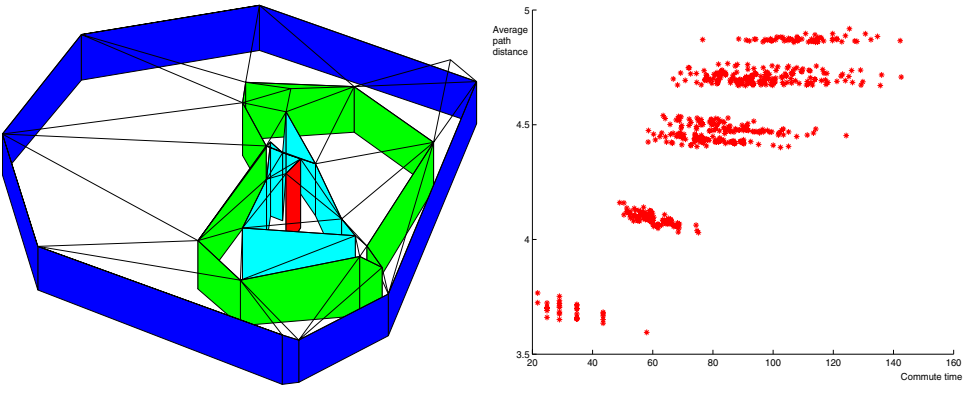
Here ϕ_1 is the eigenvector associated with eigenvalue 0 and its k-th entry is $\sqrt{d_k/vol}$. Furthermore, G can also be computed from the Laplacian spectrum using the formula $G(u,v) = \sum_{i=2}^{|V|} \frac{1}{\lambda_i} d_u^{1/2} \phi_i(u)\phi_i(v) d_v^{-1/2}$. So, the commute time is finally given by

$$O(u,v) = Q(u,v) + Q(v,u) = \frac{vol}{d_u}G(u,u) + \frac{vol}{d_v}G(v,v) - \frac{vol}{d_u}G(u,v) - \frac{vol}{d_v}G(v,u) \tag{6}$$

The score function S_u for node u is defined as $S_u = O(\tau, u)$ which is the commute time between node u and the external source node τ.

It is interesting to note that as consequence of (6) the commute time is a metric on the graph. The reason for this is that if we take the elements of G as inner products defined in a Euclidean space, O will become the norm satisfying: $\|x_i - x_j\|^2 =< x_i - x_j, x_i - x_j >=< x_i, x_i > + < x_j, x_j > - < x_i, x_j > - < x_j, x_i >$.

Figure 2(a) shows a visualisation of the score functions for the Delaunay graph 1(b). Here the score is plotted on the z-axis. The scores for the nodes on the same layer are distinct enough to separate them. In Figure 2(b) we show a scatter plot of commute times $O(u, v)$ versus the average path length distance $\mathcal{D}(u, v)$. From this plot it is clear that the commute time varies more smoothly and has a larger range.



(a) 3D visualisation of the scores on the nodes.

(b) Scatter plot of the commute time and the average path distance.

Fig. 2. 3D score visualisation and the scatter plot

5 Matching Process

Since we have divided the graph into several separate layers, our graph matching step can proceed on a layer-by-layer basis. To perform the matching process we peel layers of nodes from the boundary inwards. Each layer is a cycle graph where each node is connected to its two adjacent nodes only. In the case when a node has got only one neighbour in the layer, the edge between them is duplicated to form a cycle. We match the nodes in the corresponding layers of different graphs by performing a cyclic permutation of the nodes. The cyclic permutation permits possible null-insertions to accommodate missing or extraneous nodes. The cyclic permutation minimises the sum-of-differences in commute times between nodes in the graphs being matched. If C_l denotes the set of nodes in the kth layer of the graph, then the permutation $\mathcal{P}$ minimises the cost function

$$\mathcal{E}(\mathcal{P}) = \sum_{k} \sum_{l \in C_k^M} \sum_{m \in C_k^D} (S_l - S_{\mathcal{P}(m)})^2$$

6 Experiments

The data used in our study is furnished by a sequence of views of a model-house taken from different camera viewing directions. In order to convert the images into abstract graphs for matching, we extract point features using a corner detector. Our graphs are the Delaunay triangulations of the corner-features. Examples of the images are shown in Figure 4.

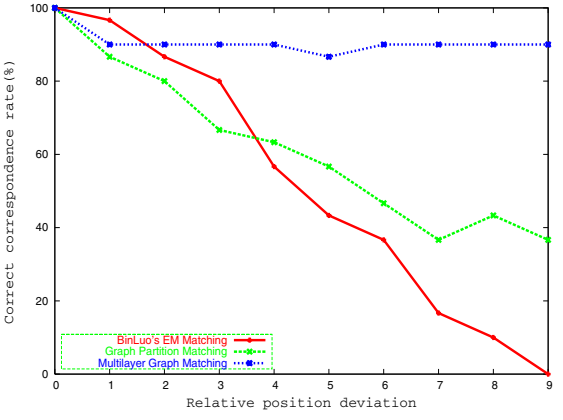
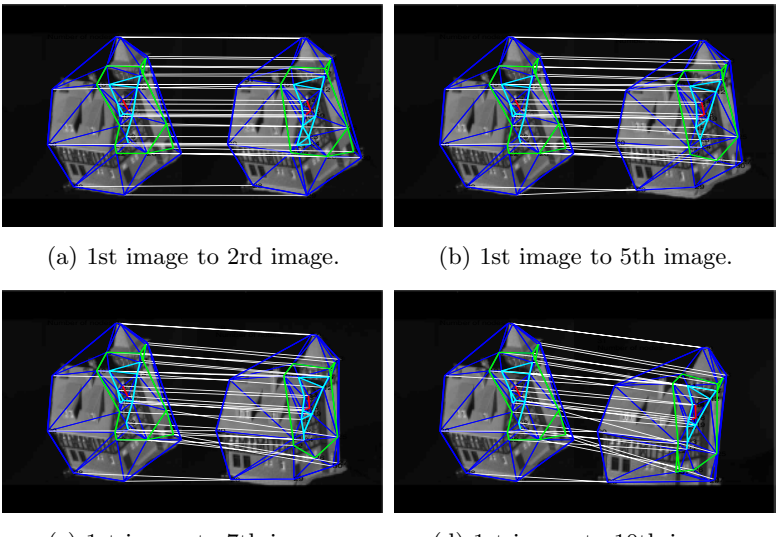


Fig. 3. Comparison of results



(a) 1st image to 2rd image. (b) 1st image to 5th image.

(c) 1st image to 7th image. (d) 1st image to 10th image.

Fig. 4. Matched samples

We have matched the first image to each of the subsequent images in the sequence by using the multilayer matching method outlined earlier in this paper. The results are compared with those obtained using the method of Luo and Hancock [7] and the partition matching method of Qiu and Hancock [9] in Table 1. This table contains the number of detected corners to be matched, the number of correct correspondences, the number of missed corners and the number of miss-matched corners. Figure 3 shows us the correct correspondence rate as a function of the difference in view number for the three methods based on the data in the Table 1.

Table 1. Correspondence allocation results and comparison with the other methods

Method	House index	0	1	2	3	4	5	6	7	8	9
	Corners	30	32	32	30	30	32	30	30	30	31
EM[7]	Correct	-	29	26	24	17	13	11	5	3	0
	False	-	0	2	3	8	11	12	15	19	24
	Missed	-	1	2	3	5	6	7	10	8	6
Partition matching[9]	Correct	-	26	24	20	19	17	14	11	13	11
	False	-	3	5	8	11	12	16	15	17	19
	Missed	-	1	1	2	0	1	0	4	0	0
Multilayer matching	Correct	-	27	27	27	27	26	27	27	27	27
	False	-	3	3	2	2	3	2	2	2	2
	Missed	-	0	0	1	1	1	1	1	1	1

From the results, it is clear that our new method out performs both Luo and Hancock's EM method and, Qiu and Hancock's partition matching method for large differences in viewing angles. Figure 4 shows the results for some examples image pairs. There are clearly significant structural differences in the graphs including rotation, scaling and perspective distortion. Even in the worst case, our method has a correct correspondence rate of 86.67%.

7 Conclusion

In this paper we have described how the commute time to an auxiliary node can be used for the purposes of graph matching. We make two uses of this attribute. First, we use it to define cycle graphs, concentric with the boundary of the graph. Second, we use it as a node attribute which is used to establish correspondences between the cycles under cyclic-permutation. We demonstrate how to compute the commute time using the discrete Green's function of the graph, and explain how this is related to the Laplacian spectrum. Our future plans involve using the commute times to embed the nodes of the graph in a low dimensional space, and to use the characteristics of the embedded node points for the purposes of graph-clustering.

References

1. S. Brin and L.Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
2. F.R.K. Chung. *Spectral Graph Theory*. CBMS series 92. American Mathmatical Society Ed., 1997.
3. F.R.K. Chung and S.-T. Yau. Discrete green's functions. In *J. Combin. Theory Ser.*, pages 191–214, 2000.
4. M. Gori, M. Maggini, and L. Sarti. Graph matching using random walks. In *ICPR04*, pages III: 394–397, 2004.
5. R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. *19th Intl. Conf. on Machine Learning (ICML) [ICM02].*, 2002.
6. Lszl Lovsz. Random walks on graphs: A survey.
7. B. Luo and E. R. Hancock. Structural graph matching using the em algorithm and singular value decomposition. *IEEE PAMI*, 23(10):1120–1136, 2001.
8. M. Meila and J. Shi. A random walks view of spectral segmentation, 2001.
9. H. Qiu and E.R. Hancock. Spectral simplification of graphs. ECCV, 2004.
10. A. Robles-Kelly and E. R. Hancock. String edit distance, random walks and graph matching. *In Int. Journal of PRAI*, 18(3):315–327, 2004.
11. M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. In *LN-AI*, 2004.
12. V. Sood, S. Redner, and D. ben Avraham. First-passage properties of the erdoscrenyi random graph. *J. Phys. A: Math. Gen.*, (38):109–123, 2005.

Theoretical and Algorithmic Framework for Hypergraph Matching

Horst Bunke[1], Peter Dickinson[2], and Miro Kraetzl[2]

[1] Institut für Informatik und angewandte Mathematik,
Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
bunke@iam.unibe.ch
[2] Intelligence Surveillance Reconnaissance Division, Defence Science and
Technology Organisation, Edinburgh SA 5111, Australia
{Peter.Dickinson, Miro.Kraetzl}@dsto.defence.gov.au

Abstract. Graphs have been successfully used in many disciplines of science and engineering. In the field of pattern recognition and image analysis, graph matching has proven to be a powerful tool. In this paper we generalize various matching tasks from graphs to the case of hypergraphs. We also discuss related algorithms for hypergraph matching.

1 Introduction

Graphs have been successfully used in many disciplines of science and engineering. Recent work in structural pattern recognition has resulted in a number of new graph matching algorithms that can be used to compute the similarity or, equivalently, the distance of a given pair of graphs. This led to many interesting applications of graph matching, including document analysis [1], shape recognition [2], biometric identification [3], and other tasks. For a recent survey that covers both methodology and applications of graph matching see [4].

Although many successful applications of graph matching have been reported in the literature, graphs are restricted in the sense that only binary relations between nodes can be represented, through graph edges. An extension is provided by hypergraphs, where each edge is a subset of the set of nodes [5]. Hence higher-order relations between nodes can be directly modeled in a hypergraph, by means of hyperedges. A large body of theoretical work on hypergraphs has been published. However, not many applications in the field of image processing and pattern recognition involving hypergraphs have been reported. Ref. [6] lists a number of hypergraph applications in low level image processing, and [7] describes a 3-D object recognition system using hypergraphs. We notice in particular that there is a lack of hypergraph matching algorithms. It seems that only the problems of maximum common sub-hypergraph [8] and hypergraph monomorphism [7] have been considered in the literature until now.

In this paper, in Section 2, we formally introduce hypergraphs as an extension of ordinary graphs. Then, in Section 3, we generalize a number of matching concepts from ordinary graphs to the case of hypergraphs. Algorithmic procedures

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 463–470, 2005.

for actually performing hypergraph matching are discussed in Section 4. Finally, Section 5 concludes this paper and provides suggestions for future research.

2 Preliminaries

Let L_V and L_E be finite sets of labels for the nodes and edges of a graph, or the nodes and hyperedges of a hypergraph, respectively.

Def. 2.1: A graph is a 4-tuple $g = (V, E, \alpha, \beta)$, where V is the finite set of nodes (also called vertices), $E \subseteq V \times V$ is the set of edges, $\alpha : V \rightarrow L_V$ is a function assigning labels to nodes, and $\beta : E \rightarrow L_E$ is a function assigning labels to edges.

The edges of a graph can be interpreted as unary or binary relations. Unary relations correspond to loops, i.e. edges $(x, x) \in E$, while binary relations correspond to directed edges of the form $(x, y) \in E$, $x \neq y$. Hypergraphs are a generalization of ordinary graphs in the sense that higher-order relations between nodes can be modeled.

Def. 2.2: Let $N \geq 1$ be an integer. A *hypergraph* of order N is a 4-tuple $h = (V, \mathcal{E}, \alpha, \mathcal{B})$, where V is the finite set of nodes (also called vertices), $\mathcal{E} = \cup_{i=1}^{N} E_i$ with $E_i \subseteq V^i$ is the set of hyperedges $(i = 1, \ldots, N)$, $\alpha : V \rightarrow L_V$ is a function assigning labels to nodes, and $\mathcal{B} = \{\beta_1, \ldots, \beta_N\}$ is the set of hyperedge labeling functions with $\beta_i : E_i \rightarrow L_E$.

Each E_i is a subset of the set of hyperedges. It consists of i-tuples $(x_1, \ldots, x_i) \in V^i$, where each i-tuple is a hyperedge of hypergraph h. We call i the order of hyperedge $e = (x_1, \ldots, x_i)$. The elements of E_1 are the loops of the hypergraph and the elements of E_2 correspond to the edges of a (normal) graph. A hyperedge of degree higher than two can be used to model higher-order relations among the nodes. Note that graphs according to Def. 2.1 are a special case of Def. 2.2 if $N = 2$.

There are several possibilities to graphically represent a hypergraph. In [5] it was proposed to draw a node as a point, an edge from subset E_1 as a loop, an edge from subset E_2 as a line segment connecting the pair of nodes involved, and edges of higher degree as simple closed curves that enclose the corresponding nodes. In this paper we adopt a different graphical notation, where circles are used to graphically represent nodes, and ellipses are used to represent hyperedges of degree three and higher. If $e = (x_1, \ldots, x_i) \in E_i$, $i \geq 3$, then we draw i line segments connecting the hyperedge symbol and $x_1, \ldots, x_i$, respectively. Labels are written next to nodes or next to hyperedge symbols.

Several hypergraph formalisms have been proposed in the literature. For a more detailed discussion of how these formalisms are related to the current paper we refer the reader to [9].

We conclude this section with a few examples that illustrate how hypergraphs can be used in pattern recognition and image analysis. These examples are also

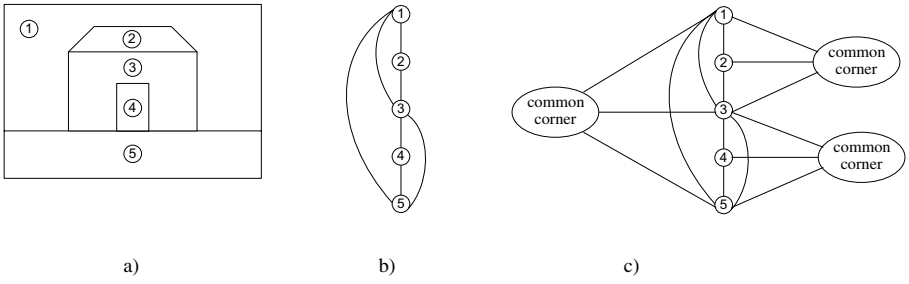


Fig. 1. a) image after segmentation; b) region adjacency graph; c) hypergraph with hyperedges representing region adjacency of order two and three

intended to show that hypergraphs have a higher representational power than normal graphs.

Example 2.1: In pattern recognition and computer vision, *region adjacency graph* (rag) is a popular datastructure to formally represent the contents of an image after it has been segmented into homogeneous regions. In a *rag* nodes represent image regions and edges between nodes indicate whether two regions are adjacent to each other or not. Fig. 1a shows an image that has been segmented into homogeneous regions, and Fig. 1b shows the corresponding *rag*. For certain applications it may be interesting to know all the regions that meet at a common corner in the image. Such a relation among three or more regions can't be directly represented in a normal graph. But in a hypergraph it is straightforward to model relations of this kind by means of hyperedges. Fig. 1c shows a hypergraph that corresponds to Fig. 1a and includes region adjacency of degree three.[1] □

Example 2.2: Wireframe models are a common representation in 3-D object modeling and computer vision. Fig. 2a shows a polyhedral object and Fig. 2b the corresponding wireframe model, given in terms of a graph. In this representation graph nodes represent the vertices of the polyhedron, and graph edges correspond to the edges of the polyhedron. Note that the graph in Fig. 2b only captures the topology of the object, and doesn't include any quantitative, metric information. In a graph, such as the one depicted in Fig. 2b, only binary relations between the vertices of an object can be represented. In a hypergraph it is easy to directly model relations of higher order, such as collinearity and coplanarity, by means of hyperedges. Fig. 2c shows an extended version of Fig. 2b, where the relation of collinearity has been added. □

3 Hypergraph Matching

Graph matching is the task of establishing a correspondence between the nodes and edges of two given graphs such that some constraints are satisfied [4]. Well-

[1] Actually there are two instances of each relation of degree 3 in the image. However, this observation is not modeled in Fig. 1c, i.e. only one instance is included.

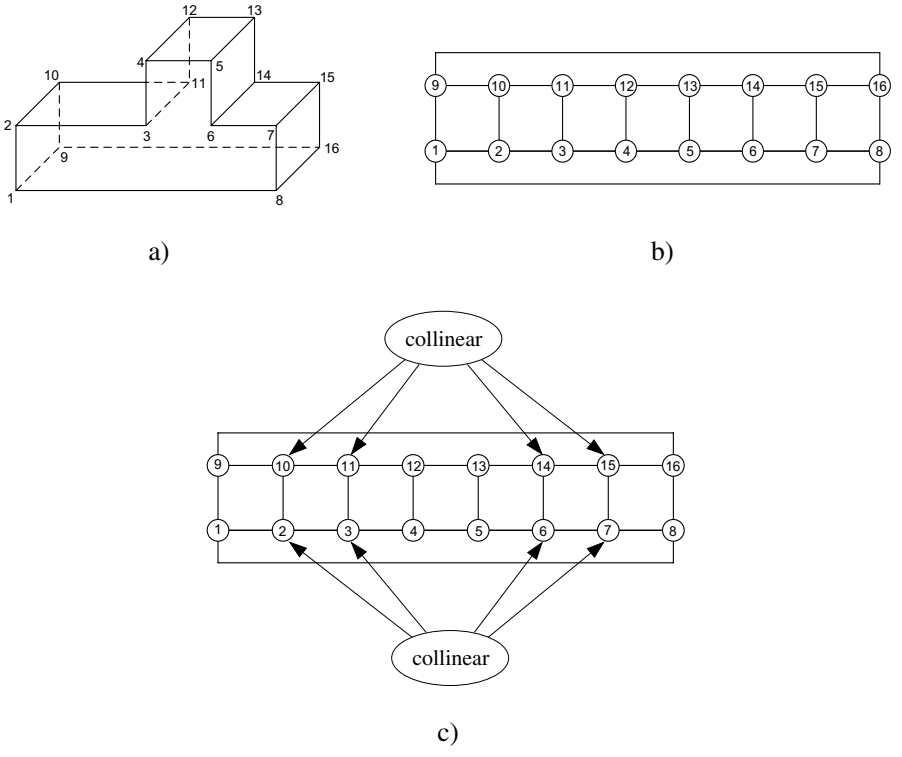


Fig. 2. a) a polyhedral object; b) graph representation; c) hypergraph showing collinear vertices

known instances of the graph matching problem include graph and subgraph isomorphism [10], maximum common subgraph computation [11] and graph edit distance [12]. In this section we'll extend these matching problems from graphs to the case of hypergraphs.

Def. 3.1: Let $h = (V, \alpha, \mathcal{E}, \mathcal{B})$ and $h' = (V', \alpha', \mathcal{E}', \mathcal{B}')$ be two hypergraphs of order N and N', respectively. We call h a *sub-hypergraph* of h' if $V \subseteq V'$; $E_i \subseteq E'_i$ for $i = 1, \ldots, N$; $\alpha(x) = \alpha'(x)$ for all $x \in V$; $\beta_i(e) = \beta'_i(e)$ for all $e \in E_i$ and $i = 1, \ldots, N$.

According to Def. 3.1 the inclusion of a hyperedge in a sub-hypergraph requires all its nodes being present as well. Otherwise the hyperedge will be dropped from the sub-hypergraph.

Def. 3.2: Let h and h' be two hypergraphs with $N = N'$. A *hypergraph isomorphism* between h and h' is a bijective mapping $f : V \to V'$ such that $\alpha(x) = \alpha'(f(x))$ for all $x \in V$; for $i = 1, \ldots, N$ and any hyperedge $e = (x_1, \ldots, x_i) \in E_i$ there exists a hyperedge $e' = (f(x_1), \ldots, f(x_i)) \in E'_i$ such that $\beta_i(e) = \beta'_i(e')$ and for any hyperedge $e' = (f(x_1), \ldots, f(x_i)) \in E'_i$ there exists a hyperedge $e = (f^{-1}(x_1), \ldots, f^{-1}(x_i)) \in E_i$ such that $\beta'_i(e') = \beta_i(e)$.

If $f : V \rightarrow V'$ is a hypergraph isomorphism between two hypergraphs, h and h', and h' is a sub-hypergraph of another hypergraph, h'', then f is called a *sub-hypergraph isomorphism* from h to h''.

Def. 3.3: Let h and h' be two hypergraphs. A *common sub-hypergraph* of h and h' is a hypergraph h'' such that there exist sub-hypergraph isomorphisms from h'' to h and from h'' to h'. We call h'' a *maximum common sub-hypergraph* of h and h', if there exists no other common sub-hypergraph of h and h' that has more nodes and, for a fixed number of nodes, more hyperedges than h''. □

Next we like to mention that it is straightforward to extend edit distance from the case of normal graphs to hypergraphs. Because of limited space we give only an informal description here. For a complete and formal treatment see [9]. Given two hypergraphs, h and h', and a set of edit operations with associated costs, one can transform h into h' by application of a suitable sequence of edit operations. The edit distance of hypergraphs h and h' is defined as the cost of the cheapest sequence of edit operations that transform h into h', i.e. $d(h, h') = min_S\{c(S) \mid S \text{ is a sequence of edit operations transforming } h \text{ into } h'\}$.

The edit distance, $d(h, h')$, measures the difference, or distance, of a pair of hypergraphs, h and h'. Clearly, if h and h' are isomorphic then $d(h, h') = 0$. In general, the greater the dissimilarity between h and h' is, the more edit operations are needed to transform h into h' and the larger the edit distance becomes.

In ordinary graph matching, a few other distance measures have been proposed. They are based on the maximum common subgraph of a pair of graphs. One of those measures [13] is defined as

$$d(g, g') = 1 - \frac{|mcs(g,g')|}{max(|g|,|g'|)} \tag{3.1}$$

In this definition, $|g|$ denotes the size of graph g, for example, the number of nodes, and $mcs(g, g')$ is the maximum common subgraph of graphs g and g'. Clearly this definition can be applied to hypergraphs as well if we replace maximum common subgraph by maximum common sub-hypergraph.

There are applications where one needs to represent a set of graphs by a single prototype. In [14] the median graph has been introduced to accomplish this task. Formally, the median of a set of graphs, $G = \{g_1, \ldots, g_k\}$, is a graph, $\overline{g}$, that satisfies

$$\sum_{i=1}^{k} d(\overline{g}, g_i) = \min\{\sum_{i=1}^{k} d(g, g_i) \mid g \in U\} \tag{3.2}$$

In this definition, U is the set of all graphs with node and edge labels from L_V and L_E, respectively. Hence the median is a graph that has, among all graphs with labels from L_V and L_E, the smallest possible average edit distance to the members of the given set, G.

Median graph computation has a very high computational complexity. However, if we restrict set U to be identical to G, we just need to compute all pairwise

distances $d(g_i, g_j)$, where $i, j = 1, \ldots, k$ and $i \neq j$, and select that graph g_i from set G that has the smallest sum of distances. Obviously, this procedure can be directly generalized from graphs to hypergraphs. We only need to replace graph distance by hypergraph distance.

4 Algorithms for Hypergraph Matching

In the previous section, a number of theoretical concepts have been introduced. However, no algorithmic procedures were considered. In the current section we'll discuss possible algorithms for hypergraph matching. For the matching of normal graphs, many algorithms have been proposed in the literature. They are based on various computational paradigms, including combinatorial search, neural networks, genetic algorithms, graph eigenvalue decomposition, and others. For a recent survey we refer to [4].

We start with the problem of extending graph and subgraph isomorphism computation to the case of hypergraphs. One of the best known graph matching algorithms, which can be used for both graph and subgraph isomorphism detection, is the one by Ullman [10]. It is a combinatorial search procedure that explores all possible mappings between the nodes of the two graphs under consideration. In order to avoid the exploration of partial mappings that can't lead to a correct solution, a look-ahead strategy is used. In this section, we'll discuss a generalization to of this algorithm to hypergraph matching. We refer again to [9] for more details.

Given two graphs, g_1 and g_2, that need to be tested for subgraph isomorphism, Ullmann's algorithm sequentially maps each node, x, of g_1 to a node, y, of g_2 and checks a number of constraints. Let $x \in V_1$, $y \in V_2$ and $f : V_1 \rightarrow V_2$ the mapping being constructed. The partial mapping constructed up to a certain point in time is augmented by $f(x) = y$ if the following three constraints are satisfied:

1. There exists no node $x' \in V_1$, $x' \neq x$, with $f(x') = y$, i.e. no other node of g_1 has already been assigned to y
2. Nodes x and y have the same label
3. The assignment $f(x) = y$ is compatible with all previous node assignments under function f, i.e. if the assignment $f(u) = v$ has been made before and there is an edge (x, u) or (u, x) in g_1, there must be an edge $(f(x), f(u))$ or $(f(u), f(x))$ in g_2 with the same label

To extend Ullmann's algorithm to the case of hypergraphs, we adopt constraints 1 and 2 without any changes. Only constraint 3 needs to be generalized in the sense that not only compatibility with respect to all edges, but w.r.t. all hyperedges is checked. Clearly such an extension is straightforward to implement.

A significant speedup in Ullmann's algorithm is achieved through the use of a lookahead technique. The basic idea is to maintain a *future match* table where all possible future node assignments are recorded. Initially, all pairs

$(x, y) \in V_1 \times V_2$ where x and y have identical node labels are possible. During the course of the search, pairs that violate Constraint 3 are eliminated. Consequently the number of assignments to be investigated in the tree search procedure is reduced. Obviously this lookahead procedure can be integrated in our subhypergraph matching schema. Assume that $x_1, \ldots, x_n \in V_1$ have been mapped to $f(x_1), \ldots, f(x_n) \in V_2$. Once a new assignment $f(x) = y$ has been made, we inspect all future (i.e. remaining) nodes $u_1, \ldots, u_m \in V_1$ and $v_1, \ldots, v_k \in V_2$ and delete any pair (u_i, v_j) from the future match table if nodes $x_1, \ldots, x_m, x, u_i \in V_1$ are part of a hyperedge in h_1 but $f(x_1), \ldots, f(x_n), f(x), v_j$ are not part of an hyperedge with the same label and order in h'.

The computational paradigm underlying Ullmann's algorithm is tree search. Also the problem of maximum common subgraph and graph edit distance computation can be solved by means of the tree search [11]. In case of maximum common subgraph computation an isomorphism between the first graph, g_1, and the second graph, g_2, is constructed that has the maximum possible size. This procedure can be extended from graphs to hypergraphs by extending the consistency checks in the common subgraph from edges to hyperedges. In case of graph edit distance computation a mapping of the nodes of g_1 to the nodes of g_2 is constructed such that the cost of the edit operations implied by this mapping is minimized. This procedure can be generalized to hypergraphs by extending edit operations on the edges to hyperedges.

5 Conclusions

Graphs has become a well-established representation formalism in pattern recognition. There is a large number of applications where graphs and graph matching algorithms have been used successfully [4]. Nevertheless, graphs are restricted in the sense that only two-dimensional relations can be modeled. In this paper we have investigated a more general framework that is based on hypergraphs.

Hypergraphs allow us to model not only binary relations, but relations of any finite order, and include graphs as a special case. A fundamental requirement for any graph-based formalism in pattern recognition is the availability of related graph matching algorithms. For the case of normal graphs, such algorithms exist. Examples are isomorphism, subgraph isomorphism, maximum common subgraph, and graph edit distance computation. On top of such algorithms, classification and clustering procedure can be implemented. In this paper we show that similar matching algorithms can be designed for the domain of hypergraphs. This makes the enhanced representational power of hypergraphs available for a potentially large number of practical pattern recognition applications.

The main purpose of this paper was to introduce a theoretical and algorithmic framework for hypergraph matching. Our future work will be concerned with practical implementations of the methods proposed in this paper. Also we plan to conduct practical experiments to study the computational behavior (time and space complexity) of the proposed algorithms and compare them to classical graph matching.

References

1. Llados, J., Marti, E., Villanueva, J.: Symbol recognition by error-tolerant subgraph matching between region adjacency graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 23-10, 2001, 1137-1143
2. Luo, B., Hancock, E.: Structural graph matching using the EM algorithm and singular value decomposition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 23-10, 2001, 1120-1136
3. Marcialis, G., Roli, F., Serrau, A.: Fusion of statistical and structural fingerprint classifiers. In Kittler, J., Nixon, M., eds.: *4th Int. Conf. Audio- and Video-Based Biometric Person Authentication*, Springer LNCS 2688, 2003, 310-317
4. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition, *Int. Journal of Pattern Recognition and Art. Intelligence*, Vol. 18, No. 3, 2004, 265-298
5. Berge, C.: Hypergraphs, North-Holland, 1989
6. Bretto, A., Cherifi, H., Aboutajdine, D.: Hypergraph imaging: an overview, Pattern Recognition, Vol. 35, No. 3, 2002, 651-658
7. Wong, A.C.K., Lu, S.W., Rioux, M.: Recognition and shape synthesis of 3-D objects based on attributed hypergraphs, *IEEE Trans. PAMI*, Vol. 11, No. 3, 1989, 279-290
8. Demko, D.: Generalization of two hypergraphs. Algorithm of calculation of the greatest sub-hypergraph common to two hypergraphs annotated by semantic information. In Jolion, J.-M., Kropatsch, W. (eds.): *Graph Based Representations in Pattern Recognition, Computing*, Supplement 12, Springer Verlag, 1998, 1-10
9. Bunke, H., Dickinson, P., Kraetzl, M.: Matching Hypergraphs, *Technical Report*, Intelligence Surveillance Reconnaissance Division, Defence Science and Technology Organisation, Edinburgh SA 5111, Australia, 2004
10. Ullman, J.R.: An algorithm for subgraph isomorphism, *Journal of ACM*, Vol. 23, 1976, 31-42
11. McGregor, J.J.: Backtrack search algorithm and the maximal common subgraph problem, *Software – Practice and Experience*, Vol. 12, 1982, 23-34
12. Messmer, B.T., Bunke, H.: A new algorithm for error-tolerant subgraph isomorphism detection, *IEEE Trans. PAMI*, Vol. 20, No. 5, 1998, 493-507
13. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph, *Pattern Recognition Letters*, Vol. 19, 1998, 255-259
14. Jiang, X., Münger, A. Bunke, H.: On median graphs: properties, algorithms, and applications, *IEEE Trans. PAMI*, Vol. 23, No. 10, 2001, 1144-1151

Geometric Characterisation of Graphs

Bai Xiao and Edwin R. Hancock

Department of Computer Science,
University of York, York Y01 5DD, UK.

Abstract. In this paper, we explore whether the geometric properties of the point distribution obtained by embedding the nodes of a graph on a manifold can be used for the purposes of graph clustering. The embedding is performed using the heat-kernel of the graph, computed by exponentiating the Laplacian eigen-system. By equating the spectral heat kernel and its Gaussian form we are able to approximate the Euclidean distance between nodes on the manifold. The difference between the geodesic and Euclidean distances can be used to compute the sectional curvatures associated with the edges of the graph. To characterise the manifold on which the graph resides, we use the normalised histogram of sectional curvatures. By performing PCA on long-vectors representing the histogram bin-contents, we construct a pattern space for sets of graphs. We apply the technique to images from the COIL database, and demonstrate that it leads to well defined graph clusters.

1 Introduction

One of the problems that arises in the manipulation of large amounts of graph data is that of characterising the topological structure of individual graphs. One of the most elegant ways of doing this is to use the spectrum of the Laplacian matrix [7,4]. For instance Shokoufandeh *et al* [10] have used topological spectra to index tree structures, Luo *et al* [3] have used the the spectrum of the adjacency matrix to construct pattern spaces for graphs, and Wilson and Hancock [9] have used algebraic graph theory to construct permutation invariant polynomials from the eigenvectors of the Laplacian matrix. One way of viewing these methods is that of constructing a low-dimensional feature-space the captures the topological structure of the graphs under study.

An interesting alternative to using topological information to characterise graphs is to embed the nodes of a graph on a manifold, and to study the geometry of this manifold. Broadly speaking there are three ways in which the problem has been addressed. First, the graph can be interpolated by a surface whose genus is determined by the number of nodes, edges and faces of the graph. Second, the graph can be interpolated by a hyperbolic surface which has the same pattern of geodesic (internode) distances as the graph [1]. Third, a manifold can be constructed whose triangulation is the simplicial complex of the graph [2].

However, recently it has been shown that there is a link between the heat kernel of a graph and the geometry of the manifold on which its nodes reside

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 471–478, 2005.

[12]. The heat kernel is the solution of the heat equation on the graph, and is found be exponentiating the Laplacian eigensystem with time. The heat kernel encapsulates the way in which information flows through the edges of the graph over time, and is closely related to the path length distribution on the graph. The graph can be viewed as residing on a manifold whose pattern of geodesic distances is characterised by the heat kernel. Moreover, differential invariants of the manifold can be computed from the heat kernel, and these in turn are related to the Laplacian eigensystem. For instance, the trace of the heat kernel [4] (or the sum of the Laplacian eigenvalues exponentiated with time) can be expanded as a rational polynomial in time, and the coefficients of the leading terms in the series are directly related to the geometry of the manifold. The leading co-efficient is the volume of the manifold, the second co-efficient is related to the Euler characteristic, and the third co-efficient to the Ricci curvature. The zeta-function (i.e. the sum of the eigenvalues raised to a non-integer power) for the Laplacian also conveys geometric information since its derivative at the origin is related to the torsion tensor. This field of study is sometimes referred to as spectral geometry [5,11].

In recent work [13], we have explored how the heat kernel can be used to embed the nodes of a graph on a manifold using a precedure similar to ISOMAP [8]. Here we have performed embedding by applying multidimensional scaling (MDS) to the elements of the heat kernel. In this paper, we aim to explore a more direct approach based on using the heat kernel to characterise the geometry of the manifold on which a graph resides. When the manifold is locally Euclidean, then the heat kernel may be approximated by a Gaussian function of the geodesic distance between nodes. By equating the spectral and Gaussian forms of the kernel, we can make estimates of the Euclidean distances between nodes. The geodesic distance is given by the floor of the path-length, and this may be computed from the Laplacian spectrum. The difference between the geodesic and Euclidean distances gauges the degree of bending of the manifold, and can be used to estimate the sectional curvatures associated with the edges of the graph. We aim to explore whether sets of graphs can be projected into a pattern space by performing principal components analysis on histograms of sectional curvatures. In this way we aim to characterise the geometry of the manifold in an implicit way.

2 Heat Kernels on Graphs

In this section, we develop a method for approximating the geodesic distance between nodes by exploiting the properties of the heat kernel. To commence, suppose that the graph under study is denoted by $G = (V, E)$ where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. Since we wish to adopt a graph-spectral approach we introduce the adjacency matrix A for the graph where the elements are

$$A(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We also construct the diagonal degree matrix D, whose elements are given by $D(u,u) = \sum_{v \in V} A(u,v)$. From the degree matrix and the adjacency matrix we construct the Laplacian matrix $L = D - A$, i.e. the degree matrix minus the adjacency matrix. The normalised Laplacian is given by $\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. The spectral decomposition of the normalised Laplacian matrix is $\hat{L} = \Phi \Lambda \Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the diagonal matrix with the ordered eigenvalues as elements and $\Phi = (\phi_1 | \phi_2 | .... | \phi_{|V|})$ is the matrix with the ordered eigenvectors as columns. Since $\hat{L}$ is symmetric and positive semi-definite, the eigenvalues of the normalised Laplacian are all positive. The eigenvector associated with the smallest non-zero eigenvector is referred to as the Fiedler-vector. We are interested in the heat equation associated with the Laplacian, i.e. $\frac{\partial h_t}{\partial t} = -\hat{L} h_t$, where h_t is the heat kernel and t is time. The heat kernel can hence be viewed as describing the flow of information across the edges of the graph with time. The rate of flow is determined by the Laplacian of the graph. The solution to the heat equation is found by exponentiating the Laplacian eigenspectrum, i.e. $h_t = \Phi \exp[-t\Lambda] \Phi^T$. When t tends to zero, then $h_t \simeq I - \hat{L}t$, i.e. the kernel depends on the local connectivity structure or topology of the graph. If, on the other hand, t is large, then $h_t \simeq \exp[-t\lambda_m] \phi_m \phi_m^T$, where λ_m is the smallest non-zero eigenvalue and ϕ_m is the associated eigenvector, i.e. the Fiedler vector. Hence, the large time behavior is governed by the global structure of the graph.

2.1 Geodesic Distance

It is interesting to note that the heat kernel is also related to the path length distribution on the graph. To show this, consider the matrix $P = I - \hat{L}$, where I is the identity matrix. The heat kernel can be rewritten as $h_t = e^{-t(I-P)}$. We can perform a McLaurin expansion on the heat-kernel to re-express it as a polynomial in t. The result of this expansion is

$$h_t = e^{-t} \left(I + tP + \frac{(tP)^2}{2!} + \frac{(tP)^3}{3!} + \cdots \right) = e^{-t} \sum_{k=0}^{\infty} P^k \frac{t^k}{k!} \tag{2}$$

The matrix P has elements

$$P(u,v) = \begin{cases} 1 & \text{if } u = v \\ \frac{1}{\sqrt{deg(u) deg(v)}} & \text{if } u \neq v \text{ and } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

As a result, we have that

$$P^k(u,v) = \sum_{S_k} \prod_{i=1}^{k} \frac{1}{\sqrt{deg(u_i) deg(u_{i+1})}} \tag{4}$$

where the walk S_k is a sequence of vertices $u_0, \cdots, u_k$ of length k such that $u_i = u_{i+1}$ or $(u_i, u_{i+1}) \in E$. Hence, $P^k(u,v)$ is the sum of weights of all walks

of length k joining nodes u and v. In terms of this quantity, the elements of the heat kernel are given by

$$h_t(u,v) = \exp[-t] \sum_{k=0}^{|V|^2} P^k(u,v) \frac{t^k}{k!} \tag{5}$$

We can find a spectral expression for the matrix P^k using the eigen-decomposition of the normalised Laplacian. Writing $P^k = (I - \hat{L})^k = \Phi(I - \Lambda)^k \Phi^T$, the element associated with the nodes u and v is

$$P^k(u,v) = \sum_{i=1}^{|V|} (1 - \lambda_i)^k \phi_i(u) \phi_i(v) \tag{6}$$

The geodesic distance between nodes, i.e. the length of the walk on the graph with the smallest number of connecting edges, can be found by searching for the smallest value of k for which $P^k(u,v)$ is non zero, i.e. $d_G(u,v) = floor_k P^k(u,v)$.

2.2 Euclidean Distance

Here we are interested in using the heat-kernel to compute an approximate Euclidean distance between nodes. This is the shortest distance between nodes in the vector space in which the manifold resides. Asymptotically with small t, on a Riemannian manifold the heat kernel can be approximated by the so-called paramatrix [11]

$$h_t(u,v) = [4\pi t]^{-\frac{n}{2}} \exp[-\frac{1}{4t} d_G(u,v)^2] \sum_{m=0}^{\infty} b_m(u,v) t^m \tag{7}$$

where $d_G(u,v)$ is the geodesic distance between the nodes u and v on the manifold, n is the dimensionality of the space (taken to be 3 in our later experiments with MDS) and $b_m(u,v)$ real-valued coefficients. When the manifold is locally Euclidean, then only the first term in the polynomial series is significant and the heat kernel is approximated by a Gaussian. Hence, to approximate the Euclidean distance between the embedded nodes we can equate the spectral and Gaussian forms for the kernel, with the result

$$d_E^2(u,v) = -4t \ln \left\{ (4\pi t)^{\frac{n}{2}} \sum_{i=1}^{|V|} \exp[-\lambda_i t] \phi_i(u) \phi_i(v) \right\} \tag{8}$$

3 Geometric Properties of the Manifold

In this paper our aim is to explore whether the geometric properties of the embedding can be used for the purposes of characterising and clustering graphs. To do this, we we can make numerical estimates of the sectional curvature between pairs of nodes. The sectional curvature is determined by the degree to

which the geodesic bends away from the Euclidean chord. Hence for a geodesic on the manifold, the sectional curvature can be estimated easily if the Euclidean and geodesic distances are known. Suppose that the geodesic can be locally approximated by a circle. Let the geodesic distance between the pair of points u and v be $d_G(u,v)$ and the corresponding Euclidean distance be $d_E(u,v)$. Further let the radius of curvature of the approximating circle be $r_s(u,v)$ and suppose that the tangent-vector to the manifold undergoes a change in direction of $2\theta_{u,v}$ as we move along a connecting circle between the two points. In terms of the angle $\theta_{u,v}$, the geodesic distance, i.e. the distance traversed along the circular arc, is $d_G(u,v) = 2r_s(u,v)\theta_{u,v}$, and as a result we have that $\theta_{u,v} = g_G(u,v)/2r_s(u,v)$. The Euclidean distance, on the other hand, is given by $d_E(u,v) = 2r_s(u,v)\sin\theta_{u,v}$, and can be approximated using the McClaurin series

$$d_E(u,v) = 2r_s(u,v)\left(\theta_{u,v} - \frac{1}{6}\theta_{u,v}^2 + ...\right)$$

Substituting for $\theta_{u,v}$ obtained from the geodesic distance, we have

$$d_E(u,v) = d_g(u,v) - \frac{d_g(u,v)^3}{24r_s^2(u,v)}$$

Solving the above equation for the radius of curvature, the sectional curvature of the geodesic connecting the nodes u and v is approximately

$$k_s(u,v) = \frac{1}{r_s(u,v)} = \frac{2\sqrt{6}(d_G(u,v) - d_E(u,v))^{\frac{1}{2}}}{d_G(u,v)^{\frac{3}{2}}} \tag{9}$$

Since for an edge of the graph, we have that $d_G(u,v) = 1$, the squared sectional curvature associated with an embedded edge is $k_e^2(u,v) = 24(1 - d_E(u,v)$. As a result we can construct the squared sectional curvature matrix $k_e^2 = 24(A + \ln[(4\pi t)^{2nt}\Phi\exp[-4\Lambda t^2]\Phi^T])$.

To characterise the geometry of the graph embedding we construct a histogram of sectional curvatures. The sectional curvatures are assigned to m bins and the normalised contents of the jth bin is denoted by $h(j)$. The feature vector for the graph is constructed from the normalised bin-contents and $\boldsymbol{B} = (h(1), h(2), ...h(m))^T$.

For the purposes of comparison, we will compare the results of using sectional curvature histograms with a number of alternative representations. The first of these is the Laplacian spectrum. Here we use the leading m eigenvalues of the normalised Laplacian $\hat{L}$ to construct the feature-vector $\boldsymbol{B} = (\lambda_1, ..., \lambda_m)^T$. We have also explored using the normalised histogram of geodesic distances computed from the path length distribution $P^k(u,v)$.

Our aim is explore the structure of a set of graphs with pattern vectors $\boldsymbol{B}_k$, $k = 1, M$ extracted using sectional curvature histograms. There are a number of ways in which the graph pattern vectors can be analysed. Here, for the sake of simplicity, we use principal components analysis (PCA). We commence

by constructing the matrix $\mathbf{S} = [\boldsymbol{B}_1|\boldsymbol{B}_2|\ldots|\boldsymbol{B}_k|\ldots|\boldsymbol{B}_M]$ with the graph feature vectors as columns. Next, we compute the covariance matrix for the elements of the feature vectors by taking the matrix product $\mathbf{C} = \mathbf{S}\mathbf{S}^T$. We extract the principal components directions by performing the eigendecomposition $\mathbf{C} = \sum_{i=1}^{M} l_i \boldsymbol{u}_i \boldsymbol{u}_i^T$ on the covariance matrix $\mathbf{C}$, where the l_i are the eigenvalues and the $\boldsymbol{u}_i$ are the eigenvectors. We use the first s leading eigenvectors (3 in practice for visualisation purposes) to represent the graphs extracted from the images. The co-ordinate system of the eigen-space is spanned by the s orthogonal vectors $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, .., \boldsymbol{u}_s)$. The individual graphs represented by the vectors $\mathbf{B}_k, k = 1, 2, \ldots, M$ can be projected onto this eigen-space using the formula $\boldsymbol{\mathcal{B}}_k = \mathbf{U}^T \boldsymbol{B}_k$. Hence each graph G_k is represented by an s-component vector $\boldsymbol{\mathcal{B}}_k$ in the eigen-space.

4 Experiments

We have applied our geometric technique to images from the COIL data-base. The data-base contains views of 3D objects under controlled viewer and lighting conditions. For each object in the data-base there are 72 equally spaced views, which are obtained as the camera circumscribes the object. We study the images from eight example objects. A sample view of each object is shown in Figure 1. For each image of each object we extract feature points using the method of [6]. We have extracted graphs from the images by computing the Voronoi tessellations of the feature-points, and constructing the region adjacency graph, i.e. the Delaunay triangulation, of the Voronoi regions. Our technique has been applied to the resulting graph structures.

In Figure 2 we show example histograms for the twenty views for four of the objects in the COIL database. Here the histograms are stacked behind each other, and are ordered by increasing view number. There are a number of conclusions that can be drawn from the histograms. First, the histograms for the same object are relatively stable with view number. Second, the histograms for

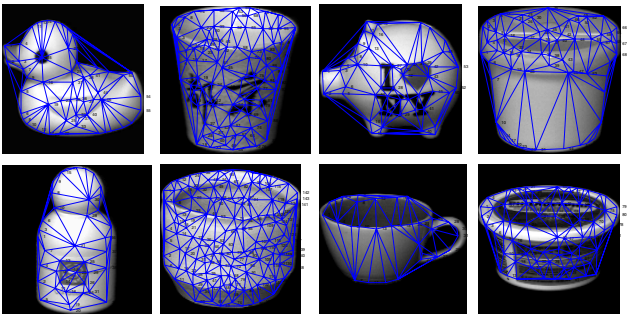


Fig. 1. Eight objects with their Delaunay graphs overlayed

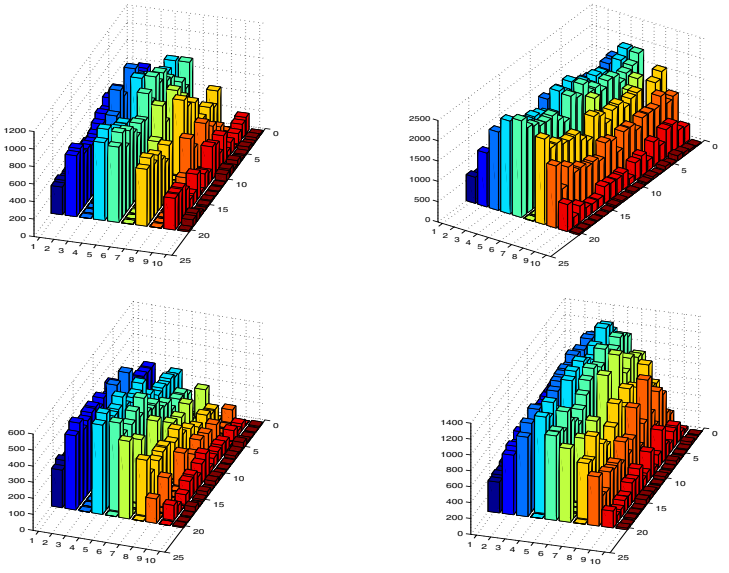


Fig. 2. Sets of histograms for different views of 4 different COIL objects

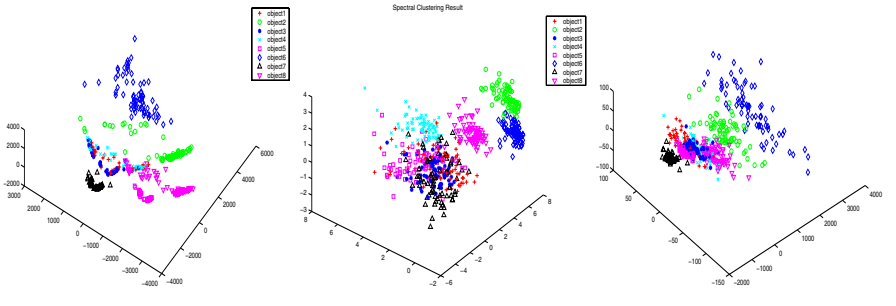


Fig. 3. Clustering results using the histograms of the sectional curvature (left), the Laplacian spectrum (centre) and the geodesic distance histogram (right).

the different objects have different shapes. The plots shown were obtained with $t = 0.001$.

The results of applying PCA to the vectors of histogram bin contents are shown in the right-hand panel of Figure 3. We obtain well defined clusters of objects. For comparison, the centre panel of Figure 3 shows the corresponding result when spectral clustering is used, and the right-hand panel that obtained using the histogram of geodesic distances. The main qualitative feature is that the different views of the ten objects are more overlapped than when the sectional curvature histograms are used. For all three clustering methods, we have computed the Rand index. For the curvature histograms the maximum rand index is 0.91 (at t=0.001), for the spectral method it is 0.57, and for the geodesic

distance histograms it is 0.72. Hence the results extracted using the curvature histograms appear to give the best clusters.

5 Conclusion and Future Work

In this paper we have explored how the use of the heat kernel can lead to a geometric characterisation of the structure of a graph. Specifically, we show how a measure of the difference between the geodesic and Euclidean distances between nodes can be used to compute sectional curvature, and this can be used for the purposes of gauging the differential structure of the manifold on which the graph resides. There are clearly a number of ways in which the work reported in this paper can be extended. For instance, it would be interesting to investigate if the distances and curvatures could be used to aid the process of visualising or drawing graphs, or for graph matching.

References

1. A.D.Alexandrov and V.A.Zalgaller. Intrinsic geometry of surfaces. *Transl.Math.Monographs*, 15, 1967.
2. A.Ranicki. Algebraic l-theory and topological manifolds. *Cambridge University Press*, 1992.
3. R. C. Wilson B. Luo and E.R. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36:2213–2223, 2003.
4. F.R.K.Chung. Spectral graph theory. *American Mathematical Society*, 1997.
5. P. B. Gilkey. Invariance theory, the heat equation, and the atiyah-singer index theorem. *Publish or Perish Inc.*, 1984.
6. C.G. Harris and M.J. Stephens. A combined corner and edge detector. *Fourth Alvey Vision Conference*, pages 147–151, 1994.
7. H.Sachs, D.M.Cvetkovic, and M.Doob. Spectra of graphs. *Academic Press*, 1980.
8. J.B.Tenenbaum, V.D.Silva, and J.C.Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:586–591, 2000.
9. B. Luo R. C. Wilson and E.R. Hancock. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:to appear, 2005.
10. A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker. Indexing using a spectral encoding of topological structure. *CVPR*, pages 491–497, 1999.
11. S.Rosenberg. The laplacian on a Riemannian manifold. *Cambridge University Press*, 2002.
12. S.T.Yau and R.M.Schoen. Differential geometry. *Science Publication*, 1988.
13. X.Bai and E.R.Hancock. Heat kernels, manifolds and graph embedding. *Structural,Syntactic, and Statistical Pattern Recognition*, pages 198–206, 2004.

Graph-Based Multiple Classifier Systems A Data Level Fusion Approach*

Michel Neuhaus and Horst Bunke

Department of Computer Science, University of Bern,
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{mneuhaus, bunke}@iam.unibe.ch

Abstract. The combination of multiple classifiers has been successful in improving classification accuracy in many pattern recognition problems. For graph matching, the fusion of classifiers is normally restricted to the decision level. In this paper we propose a novel fusion method for graph patterns. Our method detects common parts in graphs in an error-tolerant way using graph edit distance and constructs graphs representing the common parts only. In experiments, we demonstrate on two datasets that the method is able to improve the classification of graphs.

1 Introduction

The key idea in multiple classifier systems is to combine several classifiers such that the resulting combined system achieves a higher classification accuracy than the original classifiers individually [1]. In the case of statistical patterns, that is, patterns represented by feature vectors, a large number of methods for the fusion of classifiers have been developed over the past few years, ranging from the combination of individual classification results to the fusion of feature vectors. However, for structural patterns, and attributed graph patterns in particular, the fusion of classifiers has mainly been constrained to the decision level [2,3,4], that is, to the combination of the individual classifiers' results. The integration of knowledge about the structure of graphs into a multiple classifier system has not been considered until now.

In the present paper we propose a method to merge several graphs representing the same underlying pattern into a graph that is more robust against noise and hence a better structural representative of the pattern. We proceed by identifying common parts in two or more graphs and derive a graph representing the common structure. The structural matching is performed by means of graph edit distance computation. The graph fusion method can be applied when several graph representations of a pattern are given.

In Section 2, graph edit distance is briefly introduced. The proposed method for the fusion of graphs is presented in Section 3. An experimental evaluation follows in Section 4, and Section 5 offers some concluding remarks.

* Supported by the Swiss National Science Foundation NCCR program "Interactive Multimodal Information Management (IM)2" in the Individual Project "Multimedia Information Access and Content Protection".

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 479–486, 2005.

2 Graph Edit Distance

Structural pattern recognition is generally performed by transforming patterns into strings or graphs and matching the resulting structures. Using an attributed graph representation of patterns allows for a powerful representation of complex strucured objects that is better suited to certain pattern recognition problems than a statistical feature-based approach. In recent years a large number of approaches to graph matching have been proposed [5].

One of the most common error-tolerant graph matching methods is based on graph edit distance [6]. Graph edit distance is a general dissimilarity measure on attributed graphs. The key idea of edit distance is to define the dissimilarity of graphs by the amount of edit operations, reflecting small structural distortions, needed to transform one graph into another. To allow for graph distance measures that are tailored to specific applications, it is common to define for each edit operation an edit cost reflecting the strength of the corresponding structural distortion. From these edit costs, the edit distance of two graphs can be defined by the minimum cost sequence of edit operations transforming one graph into the other.

The result of an edit distance computation is a minimum cost edit path from the first to the second graph and its associated edit costs. Nodes and edges of the first graph that are substituted by nodes and edges of the second graph, according to the optimal edit path, can be regarded as locally corresponding parts of the two graphs. Conversely, inserted and deleted nodes and edges can be seen as the non-matching parts. In traditional graph matching methods, the edit distance value is used in the context of a nearest-neighbor classifier, while the optimal node and edge correspondences given by the edit path are not taken into account any further. In this paper, we propose to use the substitutions of the optimal edit path for an error-tolerant detection of the common parts of two or more graphs. The following section describes how several graph patterns can be merged into a single graph by means of edit distance.

3 Data Level Fusion of Graphs

Multiple classifier systems have successfully been used to improve graph matching systems [2,3,4]. Graph classifier fusion, however, is usually performed at the decision level. That is, each single classifier votes for a single class or reports a confidence measure for each class, and all votes or confidence measures are then combined into an overall classification result. The fusion approach we propose in this paper is based on graph fusion at the data level. We assume that each pattern is initially represented by several graphs. In practice, this is the case when several graph extraction methods have been developed based on the same key pattern characteristics, or when the same graph extraction process is carried out several times with different parameters. A crucial requirement of our method is that all those graph representations are compatible, meaning that the same attributes are used in all graphs. The basic idea is to detect common

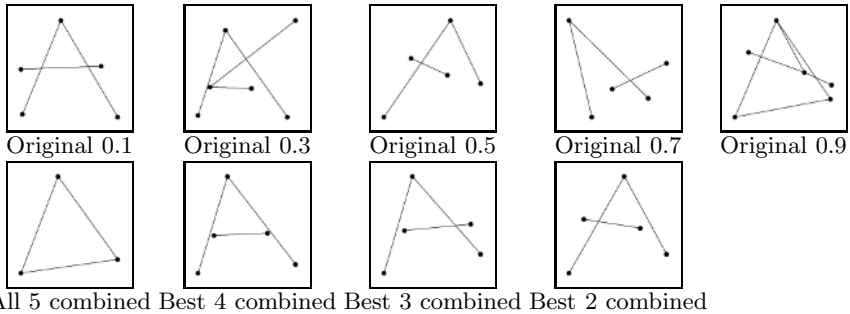


Fig. 1. Graphs from five databases (Original 0.1–0.9) and their *emcs* combinations consisting of all five graphs (All 5) as well as the best matching graphs only (Best 4–2)

structures occurring in all graphs representing a single pattern, and use these common parts in the successive matching process. The common parts typically correspond to structures that are robust against noise and distortions and thus allow for a more reliable classification.

In order to detect common structures, we use the concept of maximum common subgraph. The maximum common subgraph of two graphs is defined as the largest parts of two graphs that are isomorphic to each other [7]. To detect the common parts of two graphs in an error-tolerant way, we first compute the edit distance of the two graphs and use the substitutions of the optimal edit path as a description of the common parts. We then proceed by merging the nodes and edges of the common parts to obtain a new graph representing the structure existing in both graphs. We call this graph the error-tolerant maximum common subgraph (*emcs* graph) of the two graphs. Note that maximum common subgraph, as described in [7], is a special case of *emcs* under the condition that only identical substitutions occur.

If more than two graphs are to be merged, we compute all mutual distances and merge the two graphs with the smallest distance first. We then proceed by merging the current *emcs* graph and the remaining graphs until all graphs have been merged into a single *emcs* graph. The motivation for this procedure arises from the observation that two very different graphs will most likely lead to a small or even an empty *emcs* graph, while two very similar graphs will lead to a large graph that represents the common parts of the graphs very well. This merging procedure also has the advantage that the *emcs* graph computation can be stopped at different stages of the merging process, for instance using the *emcs* graph of the two most similar graphs only instead of the *emcs* graph of all graphs, therefore eliminating the effect of outlier graphs. The result of *emcs* merging is illustrated in Fig. 1. First, five graph instances of a letter *A* line drawing with various degrees of distortion are given. The next graph shows the complete *emcs* graph consisting of all five graphs, and the remaining graphs correspond to the *emcs* graph of the four, three, and two most similar graphs. The original graphs (see the upper row of Fig. 1) exhibit quite a significant amount of distortion in terms of added and displaced nodes and edges, while the

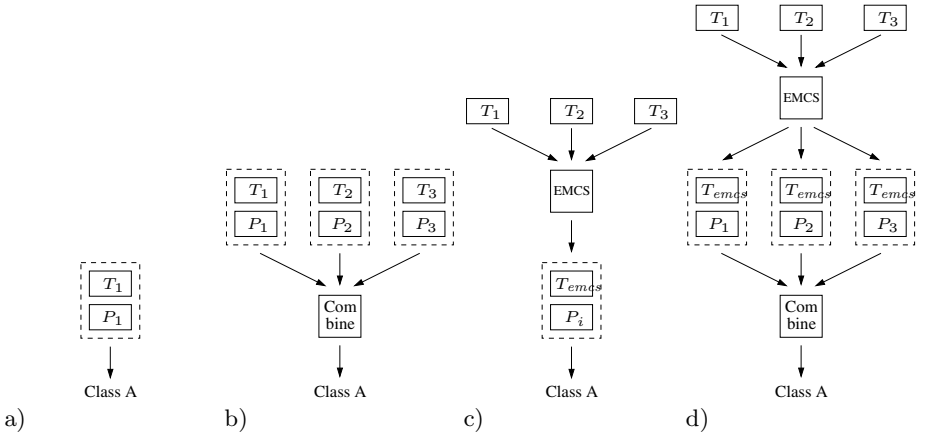


Fig. 2. a) Individual classifier, b) decision level classifier fusion, c) data level pattern fusion, and d) combination of data level and decision level fusion

emcs graphs (see the lower row of Fig. 1) constitute an intuitively less distorted and hence better representation of letter A. The *emcs* graphs indicate that it may be advantegeous to merge more than only the two best fitting graphs (Best 2), but less than all five (All 5) to benefit from a robust representation while eliminating the influence of outlier graphs. Generally, if too many graphs are taken into account in the *emcs* computation, there may be a severe influence of outliers. On the other hand, if too few graphs are considered, the variation of the underlying population may not be covered well enough.

The strategies for the fusion of graph classifiers employed in this paper are illustrated in Fig. 2b–d. To explain our notation, we first give an illustration of a single classifier of the nearest-neighbor type in Fig. 2a, consisting of a set of labeled prototype graphs P_1. In a first step, the original patterns are converted into graphs. The set of graphs to be classified, or test set, is denoted by T_1. Each graph from test set T_1 is classified according to the class of its nearest-neighbor prototype in P_1. In Fig. 2b, the fusion of classifiers at the decision level is illustrated. Here, three different graph extraction procedures are used, which means that every original pattern is represented by three different graphs, one graph in T_1, one in T_2, and one in T_3. For each of the three test sets T_1, T_2, and T_3, a corresponding set of prototype graphs is developed, denoted by P_1, P_2, and P_3, respectively. Each one of the three graphs a pattern is represented by is first individually classified with the respective nearest-neighbor classifier. To obtain an overall classification, the individual results are combined at the class level.

In Fig. 2c, the fusion is performed at the data level. Before classification, the three graphs representing the same pattern — one from T_1, one from T_2, and one from T_3 — are merged into one *emcs* graph. The module denoted by EMCS represents the error-tolerant maximum common subgraph computation, and the test set T_{emcs} represents the new set of *emcs* graphs. In this scenario, one unmodified set of prototypes, say P_i, is chosen for the nearest-neighbor clas-

sification; $i \in \{1, 2, 3\}$. That is, every *emcs* graph in T_{emcs} is classified according to the class of its nearest-neighbor prototype graph in P_i. Finally, in Fig. 2d, a combination of data level fusion and decision level fusion is shown. Again, the three test sets T_1, T_2, and T_3 are merged into a single test set T_{emcs}. Every *emcs* graph is then individually classified according to the original prototypes, that is, once according to P_1, once according to P_2, and once according to P_3. The final classification is obtained by combining the results of the individual classifiers.

4 Experimental Results

In our first experiment we focus on the data level fusion of graphs (see Fig. 2c) using an artificially created database of letter drawings. An experimental evaluation of all fusion strategies illustrated in Fig. 2, applied to the difficult problem of fingerprint classification using real-world data, is described subsequently.

In the first experiment, a clean prototype drawing is manually constructed for 15 capital letters consisting of straight lines only. These prototype drawings are then repeatedly distorted to obtain sample drawings. To control the strength of applied distortions, a distortion parameter is used. By means of this procedure, we create five databases containing 1,500 distorted drawings each, with the distortion parameter ranging from 0.1 to 0.9. A sample drawing of a letter *A* from each database is shown in Fig. 1 (Original 0.1-0.9). The letter drawings are finally transformed into attributed graphs by representing line endings by nodes (containing a position attribute) and lines by edges (without attributes), resulting in graph datasets L_1, L_2, L_3, L_4, and L_5. Each dataset L_i, $i \in \{1, \ldots, 5\}$, is split into two subsets. The test set T_i is defined as the first half of dataset L_i, and the prototype set P_i is defined as the second half of dataset L_i. Using the previously described procedure, we obtain a merged test set T_{emcs} consisting of *emcs* graphs, each of which results from merging five graphs. Similarly, a merged prototype set P_{emcs} is obtained from the prototype sets $P_1, \ldots, P_5$. The results obtained with a nearest-neighbor classifier on each test set $T_1, \ldots, T_5$ individually according to Fig. 2a are shown in the first five rows of Table 1. Using the *emcs* method according to Fig. 2c with prototype set P_{emcs} results in an improvement of the classification rate of almost 10% compared to the best indi-

Table 1. Performance of a nearest-neighbor classifier on five letter graph datasets (T_1–T_5) and the corresponding combined *emcs* dataset (T_{emcs})

Test set	Prototype set	Classification rate
T_1	P_1	54.8
T_2	P_2	54.933
T_3	P_3	51.733
T_4	P_4	70.0
T_5	P_5	76.0
T_{emcs}	P_{emcs}	85.067

Table 2. Performance of five individual fingerprint classifiers and classifier combinations using a) majority voting, b) maximum confidence, and c) confidence voting

Test set	Prototype set	Classification rate	
T_1	P_1	80.85	
T_2	P_2	74.6	
T_3	P_3	73.4	
T_4	P_4	75.5	
T_5	P_5	74.625	
$T_1, \ldots, T_5$	$P_1, \ldots, P_5$	77.35	a)
$T_1, \ldots, T_5$	$P_1, \ldots, P_5$	86.8	b)
$T_1, \ldots, T_5$	$P_1, \ldots, P_5$	83.2	c)

vidual classifier. In this particular case, we find that the *emcs* graphs constitute a representation of the line drawing patterns that is more robust against noise than the original patterns.

We proceed by investigating the applicability of the *emcs* method to fingerprint classification [8]. The method we use extracts graphs from fingerprint images by detecting characteristic signatures in fingerprints and converting these into attributed graphs [9]. Our experiments are based on five graph fingerprint classifiers using similar graph extraction procedures with different parameters. Hence, by employing five slightly different graph extraction procedures, we obtain five different graph representations for each fingerprint image. The five graph datasets $T_1, \ldots, T_5$ have been constructed from the NIST-4 database of fingerprints consisting of 4,000 fingerprint images classified according to the Henry system [8]. In a manual construction process, typically around 30 graph prototypes have been developed for each of the five datasets. The set of graph prototypes belonging to test set T_i is denoted by P_i. The fingerprint graphs from T_i are then classified according to the nearest-neighbor among all prototypes in P_i.

To obtain a measure of the reliability of each classification result, we introduce a confidence measure for distance based nearest-neighbor classifiers. The confidence measure is defined as the ratio of the distance to the nearest neighbor in relation to the distance to the nearest neighbor of the second-closest class. For a single input fingerprint, we thus obtain per classifier the resulting class along with a confidence measure. The results of the five classifiers can then be combined with majority voting (in this case the confidence values are not needed), by selecting the result of the maximum confidence classifier, or another combination rule [10].

The results of the five individual classifiers and combinations at the decision level are given in Table 2. Note that the first five rows correspond to the individual classifier scheme illustrated in Fig. 2a, and the last three rows correspond to the decision level fusion scenario illustrated in Fig. 2b. From the results, we find that among the traditional decision level combination schemes the maximum confidence rule is very effective in improving the classification accuracy. Next, we merge the five databases T_1, T_2, T_3, T_4, and T_5 into an *emcs* database T_{emcs}. The performance of the individual classifiers and the decision

Table 3. Performance of fingerprint classifier combinations using *emcs* data fusion and a) majority voting, b) maximum confidence, and c) confidence voting

Test set	Prototype set	All 5	Best 4	Best 3	Best 2	
T_{emcs}	P_1	79.625	79.925	80.75	80.45	
T_{emcs}	P_2	79.325	79.6	80.375	79.925	
T_{emcs}	P_3	78.15	78.575	79.225	78.825	
T_{emcs}	P_4	63.2	63.25	63.75	63.55	
T_{emcs}	P_5	64.725	65.15	65.625	65.925	
$T_{emcs}, \ldots, T_{emcs}$	$P_1, \ldots, P_5$	73.275	74.725	74.85	75.2	a)
$T_{emcs}, \ldots, T_{emcs}$	$P_1, \ldots, P_5$	81.45	87.3	88.275	87.8	b)
$T_{emcs}, \ldots, T_{emcs}$	$P_1, \ldots, P_5$	81.35	82.35	82.575	82.8	c)

level fusion using T_{emcs} is presented in Table 3. Note that the first five rows correspond to the data level fusion illustrated in Fig. 2c, and the last three rows correspond to the combination of data level and decision level fusion illustrated in Fig. 2d. Comparing the first five rows in Table 3 with the first five rows in Table 2, we observe that the individual classifiers on T_{emcs} do not generally lead to an improvement compared to the original classifiers. Using a fusion at the decision level, however, outperforms all other classification rates, as can be seen in rows six to eight in Table 3. Merging only two or three graphs instead of all five graphs leads to an additional gain in recognition accuracy. Again, the maximum confidence rule turns out to be the most reliable combination method.

For the sake of convenience, a summary of the experimental results is provided in Table 4. Since many fingerprint classification systems are evaluated on the second half of the NIST-4 database only instead of the full database, the recognition accuracy on the second half of NIST-4 is also given. The *emcs* data level fusion is particularly suitable in conjunction with the decision level fusion using the maximum confidence rule (see Fig. 2d). This fusion strategy is significantly better ($\alpha = 0.01$) than all individual classifiers and all other fusion strategies. We conclude that the *emcs* fusion of graph structures can improve nearest-neighbor based graph classifiers and outperform traditional fusion methods.

Table 4. Summary of classification rates obtained on NIST-4 database

Fingerprint classifier	NIST-4	Second half of NIST-4	Method
Best individual classifier	80.85	80.25	Fig. 2a
Best decision level fusion	86.8	86.95	Fig. 2b
Best data level fusion	80.75	80.55	Fig. 2c
Best data and decision level fusion	88.275	88.8	Fig. 2d

5 Conclusions

In the present paper we propose a method for data level fusion of graph patterns. Given two graphs to be merged, we first identify corresponding substructures in both graphs in an error-tolerant manner using graph edit distance. These local correspondences can then be used to construct a graph representing the common parts of the two original graphs. The merged graphs constitute robust representatives of the original graphs that are less prone to noise. By generalizing the concept to more than two graphs, we can use the graph fusion method to combine several datasets of the same underlying data into a single graph dataset. To demonstrate the usefulness of the proposed method, we apply the fusion method to artifically created line drawing graphs and the difficult problem of fingerprint classification. In both cases a significant improvement of the performance is obtained. In the future, we would like to investigate in greater detail what properties cost functions need to exhibit to be suitable for the graph fusion process.

References

1. Roli, F., Kittler, J., Windeatt, T., eds.: Multiple Classifier Systems, Proc. 5th Int. Workshop. LNCS 3077. Springer (2004)
2. Marcialis, G., Roli, F., Serrau, A.: Fusion of statistical and structural fingerprint classifiers. In Kittler, J., Nixon, M., eds.: 4th Int. Conf. Audio- and Video-Based Biometric Person Authentication. LNCS 2688 (2003) 310–317
3. Yao, Y., Marcialis, G., Pontil, M., Frasconi, P., Roli, F.: Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines. Pattern Recognition **36** (2003) 397–406
4. Schenker, A., Bunke, H., Last, M., Kandel, A.: Building graph-based classifier ensembles by random node selection. In Roli, F., Kittler, J., Windeatt, T., eds.: Proc. 5th Int. Workshop on Multiple Classifier Systems. LNCS 3077, Springer (2004) 214–222
5. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Int. Journal of Pattern Recognition and Artificial Intelligence **18** (2004) 265–298
6. Sanfeliu, A., Fu, K.: A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, Man, and Cybernetics **13** (1983) 353–363
7. Fernandez, M.L., Valiente, G.: A graph distance metric combining maximum common subgraph and minimum common supergraph. Pattern Recognition Letters **22** (2001) 753–758
8. Maltoni, D., Maio, D., Jain, A., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer (2003)
9. Neuhaus, M., Bunke, H.: A graph matching based approach to fingerprint classification using directional variance (2005) Submitted.
10. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 226–239

Improved Face Shape Recovery and Re-illumination Using Convexity Constraints

Mario Castelán* and Edwin R. Hancock

Dept. of Computer Science, University of York,
York YO10 5DD, United Kingdom
{mario,erh}@cs.york.ac.uk

Abstract. This paper describes work aimed at developing a practical scheme for face analysis using shape-from-shading. Existing methods have a tendency to recover surfaces in which convex features such as the nose are imploded. This is a result of the fact that subtle changes in the elements of the field of surface normals can cause significant changes in the corresponding integrated surface. To overcome this problem, in this paper we describe a local-shape based method for imposing convexity constraints. We show how to modifying the orientations in the surface gradient field using critical points on the surface and local shape indicators. The method is applied to both surface height recovery and face re-illumination, resulting in a clear improvement.

1 Introduction

The problem of acquiring surface models of faces is an important one with potentially significant applications in biometrics, computer games and production graphics. Shape-from-shading is one of the most appealing methods, since this is a non-invasive process which mimics the capabilities of the human vision system.

In brief, SFS aims to solve the image irradiance equation, $E(x, y) = R(p, q, s)$, where E is the intensity value of the pixel with position (x, y), R is a function referred to as *the reflectance map* [6]. The reflectance map uses the surface gradients $p = \frac{\partial Z(x,y)}{\partial x}$ and $q = \frac{\partial Z(x,y)}{\partial y}$ together with the light source direction vector s to compute a brightness estimate which can be compared with the observed one using a measure of error. If the surface normal at the location (x, y) is $n = (p, q, -1)$, then under Lambertian reflectance model, the image irradiance equation becomes $E(x, y) = n \cdot s$. In other words, the SFS problem is the one of recovering the surface that, after interaction with the environment (illumination conditions, reflectance properties of the object, inter-reflections) produces the radiances perceived by human eyes as intensities.

In general, though, SFS is an under-constrained problem since the two degrees of freedom for surface orientation (slant and tilt), must be recovered from

* Supported by National Council of Science and Technology (CONACYT), Mexico, under grant No. 141485.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 487–494, 2005.

a single measured intensity value. Hence, it is frequently posed as that of minimizing cost functionals that capture constraints on the gradient field[6]. This is usually carried out through iterative schemes in the discrete domain. Despite sustained research activity in the field for some three decades, no SFS scheme has been demonstrated to work as accurately as the *specially* constrained SFS cases of photometric stereo[4] and statistical SFS[1]. The first of these methods requires at least three images of the same object illuminated from different viewpoints, while the second uses an aligned database of accurate surface information belonging to objects of the same class, i.e. faces.

When it comes to the original single-image problem, the restrictions imposed by most SFS schemes on the gradient field (smoothness, irradiance, integrability, unit length) are insufficient to overcome with these problems, generating errors that, if propagated iteratively, can result in miscalculations on the topography of the recovered surface. For instance, because of the concave-convex ambiguity, there may be regions where the sign of the curvature becomes reversed. As far as the problem of face reconstruction is concerned, for instance, the effect can be to cause high curvature features such as the nose to become imploded with respect to the remainder of the surface. As we will demonstrate in this paper, if such regions can be identified and the surface normal directions corrected, then the result is improved overall surface topography.

For face analysis, the use of SFS has proved to be an elusive task, since the concave-convex ambiguity can result in the inversion of important features such as the nose. To overcome this problem, domain specific constraints have been used. Dogvard and Basri [3] have exploited facial symmetry. Prados and Faugeras [8] have recently proposed a SFS scheme that, under the existence of a unique critical point (i.e. the brightest intensity appearing at only one location in the image) and a light source at the optical center of the camera (rather than a homogeneous horizontal light) gives qualitatively good facial reconstructions.

It is clear that a precise height map is difficult to obtain by integration of the field of surface normals delivered by SFS due to local errors in direction and potential reversal in the sign of the curvature. However, this does not imply that the entire field of surface normals is in error.

The outline of the this paper is as follows. First we review the concepts of local shape indicators. With this knowledge at hand we develop a method that can be used to reassign the surface gradient orientations. We provide experiments to evaluate the method on height recovery and re-illumination using single images of human faces. Finally, we present some conclusions and identify directions for future work.

2 Local Shape Indicators

Curvature attributes have been widely used in shape analysis, especially for segmentation and recognition. A local shape indicator is an scalar that conveys information concerning the local topography of a surface using its principal curvatures. The principal curvatures (κ_1 and κ_2) are the eigenvectors of the Hessian

matrix, which can be computed through local changes in the surface normal directions.

Local shape indicators are usually coupled. For instance, the HK classification [2] uses the Gaussian and mean curvatures $H = \frac{(\kappa_1+\kappa_2)}{2}$ and $K = \kappa_1 * \kappa_2$ respectively. By distinguishing between the cases in which H and K are individually negative, zero or positive, it is possible on the basis of their joint behavior to assign topographic labels to points on a surface. A different and slightly more convenient set of attributes is the curvedness/shape-index representation developed by Koenderink and Van Doorn[7]. Here the principal curvatures are used to compute the shape index $S = -\frac{2}{\pi} arctan(\frac{\kappa_1+\kappa_2}{\kappa_1-\kappa_2})$ for $\kappa_1 \geq \kappa_2$; and the curvedness, $C = (\frac{\kappa_1^2+\kappa_2^2}{2})^{1/2}$.

The shape index is an angular variable that relates directly to the local surface topography. It varies continuously from -1 to $+1$ as the surface changes through cup, rut, saddle-rut, saddle, saddle-ridge, ridge and dome, and cup again. The curvedness is simply the degree of curvature of the surface. The curvedness is a convenient indicator of potential surface discontinuity. The higher the curvedness, the more likely the presence of a rapid variation in height. For instance, in the case of faces, the curvedness is large for features such as the boundaries of the nose, mouth and eyes.

For our experiments, we utilize the local descriptors of shape-index and curvedness to characterize the regions on a gradient field where a change of orientation should be performed.

3 Using Local Shape Indicators to Redirect SFS Gradient Fields

Inevitably, any surface gradient field delivered by SFS will be inaccurate due to noise or albedo changes which cause variations in the intensities of the input image. SFS works well for objects that are uniformly concave or convex. However, if the object under study is more complex, with both concave and convex regions, then SFS can fail. In these situations although the recovered surface normal direction is consistent with the measured image brightness, the recovered surface does not reflect the structure of the object under study. In particular, there may be inversions of the sign of the surface curvature with convex regions appearing concave and vice-versa. However, in the case of faces (and many other objects) the surface under study is largely convex (with the exception of the eye-sockets, the bridge of the nose and areas around the lips). Based on this observation, we present a method for enforcing the convexity of the integrated surface while ensuring a global maximum on a particular position on it.

Formally stated, suppose S is a smooth surface immersed in $\mathbb{R}^3$. Let p and U_p be a critical point of S and a neighborhood of p respectively. Suppose that S is locally concave over U_p. Then, the new surface $\widetilde{S}$ constructed from S by reversing the sign of all its partial derivatives, S_x and S_y, is locally convex in U_p. Besides, a local maximum on $\widetilde{S}$ will be located at that point where the function ceases

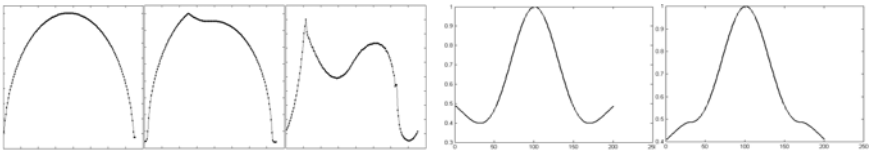


Fig. 1. Applying the method to a sphere and mexican hat(see text)

increasing and commences decreasing[1]. Suppose that all the partial derivatives of $\widetilde{S}$ with respect to x, $\widetilde{S}_x$, have negative sign before reaching the position of the critical point p on the x axis and have positive sign after reaching it, and that the same conditions hold for $\widetilde{S}_y$. The critical point p on U_p will be the position of the global maximum[2] of $\widetilde{S}$.

The basic idea underlying this paper is to enforce the condition that the integrated surface has a global height maximum at the tip of the nose. Such a point will serve as a division for the positive and negatively signed areas of the needle map. To enforce this condition we apply the simple rule:

$$\breve{Z}_x(x,y) = \begin{cases} abs(\hat{Z}_x(x,y)) & \text{if } x \leq a \text{ and } ShI(x,y) \geq \tau_x \\ -abs(\hat{Z}_x(x,y)) & \text{if } x > a \text{ and } ShI(x,y) \geq \tau_x \\ \hat{Z}_x(x,y) & \text{otherwise} \end{cases}$$

$$\breve{Z}_y(x,y) = \begin{cases} abs(\hat{Z}_y(x,y)) & \text{if } y \leq b \text{ and } ShI(x,y) \geq \tau_y \\ -abs(\hat{Z}_y(x,y)) & \text{if } y > b \text{ and } ShI(x,y) \geq \tau_y \\ \hat{Z}_y(x,y) & \text{otherwise} \end{cases}$$

where $\breve{Z}_x$ and $\breve{Z}_y$ are the updated gradients, $\hat{Z}_x$ and $\hat{Z}_y$ are the original gradients, a and b are the coordinates for the position of the desired global height maximum, on the x and y axis respectively. A local shape indicator ShI, which could be either the curvedness, the shape index or the mean and gaussian curvatures, is compared to the thresholds τ_x and τ_y for deciding whether the element of the gradient field at the location (x, y) will be altered or not.

The following diagrams show the behavior of the method when applied to the derivatives of a sphere and a mexican hat. Both experiments were realized using the curvedness indicator.

To illustrate the global height maximum enforcement procedure, Figure 1 (left) shows the results of applying the method to the derivatives of a sphere with radius 75 units. From left to right, transversal plots of the recovered surface on the x axis are shown. The peak coordinates (a, b) are both set to 75, 50 and 20 respectively, and τ_x and τ_y are set to zero: all the derivatives are taken into

[1] Of course, $\widetilde{S}$ will present many local maxima for a face-like surface.
[2] It might be a maximum or a minimum depending on the integration method.

account to expose the extreme case (though for a sphere it does not make sense to set τ different to zero). The convexity strengthening is clearer in Figure 1 (right), where the method is applied to the mexican hat function. Cross sections of the recovered surface are shown, and from left to right they show the original surface and recovered surface after applying the method taking as a peak point the center of the surface with $\tau_x = \tau_y = 0$. Note how the concave parts of the hat become convex.

In the following section some experiments will be presented in order to illustrate these points on an application involving face reconstruction using SFS.

4 Experiments

This section is organized into two parts. The first part describes experiments focussed on height recovery while the second part describes re-illumination tests.

The first series of tests were carried out on the image shown in Figure 2 (top-left)[3]. To compute the surface gradients from the raw image brightness we followed the procedure described in[9]. This construction ensures the image irradiance equation to be satisfied. For the surface integration step we used the global method proposed in [5] which recovers surface height using the inverse Fourier transform of the field of surface normal directions.

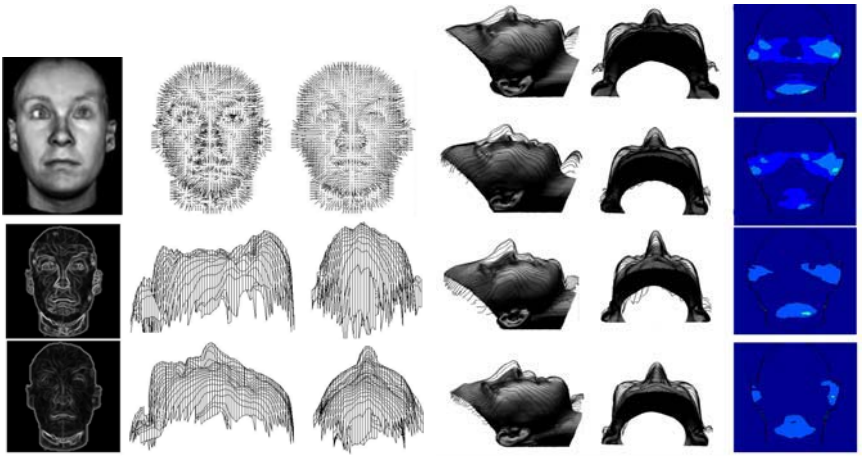


Fig. 2. Height and gradient analysis (see text)

In Figure 2 (three first columns), the first row shows the input image, the initial needle map and the curvedness-modified ($\tau_x = 0.3$ and $\tau_y = 0.4$) needle map. It is worth mentioning that, for the hardest case, where no data is filtered by the thresholds, a four-quadrant effect is quite perceptible, but it tends to

[3] The face database was provided by the Max-Planck Institute for Biological Cybernetics in Tuebingen, Germany

relax when the filter starts to work. The harder the threshold, the stronger the influence of the face features. On the other hand, the middle and bottom rows show, from left to right the curvedness map, together with profile and top-down views of the surface wire-frames recovered using surface integration. The next row is for $\tau_x = 0.3$ and $\tau_y = 0.4$, respectively. It is important to highlight how the change of signs in the surface normals suggests that the new derivatives arise now from the *shape* of a face (bottom row) rather than from the *image* of a face (middle row). By incrementing τ_x and τ_y the fine features of the face seem to be preserved and the overall structure of the face is still sound. By choosing an appropriate threshold we are able to enhance the salient features of a particular face while maintaining the overall face composition.

As far as the reconstruction using shape-index is concerned, the height data recovered is very similar to that obtained using curvedness. Figure 3 (left) shows a scatter plot comparison between the original and modified shape-indexes[4]: the x-axis corresponds to the original case while the y-axis corresponds to the$\tau_x = \tau_y = 0.4$ case[5]. This diagram presents only those pixels in the original gradient field with a shape index lower or equal to 0.4 (x-axis), therefore we can analyze the new value of such pixels in the redirected field of surface normals (y-axis). Note how the majority of the points are distributed above the line $x = y$, which shows how the original shape index turned into a grater one, suggesting that the concave regions changed to convex. Such pixels belong to the regions surrounding the nose, mouth and eyes. The small cloud of points below the line, representing the pixels where the shape index remained lower than the threshold is mainly related to the pixels surrounding the face boundary.

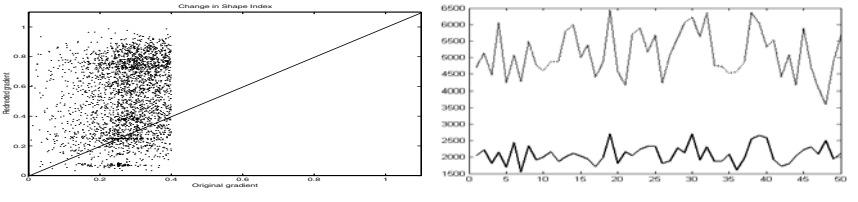


Fig. 3. Convexity enforcement using shape index and height difference tests (see text)

The images shown in Figure 2 (three last columns) provides an absolute height difference analysis. The left-most column shows the color-coded iso-contour plot of the absolute height differences after re-directing the surface normals. The middle and right-most columns show superimposed plots of the recovered height maps after applying our method on the profile views of the images from which the original gradient fields were calculated. We can observe from the iso-contour images that the absolute height difference is diminished after changing the direction of the surface normals. Errors are most significant in the chin and eye areas. This might be a consequence of change of albedo (eye

[4] The shape-index scale was normalized from 0 to 1.

[5] For all subsequent experiments, both τ_x and τ_y will slightly fluctuate between 0.3 and 0.4

area) and instabilities produced by the boundaries of the chin and neck. From the superimposed, it is clear that the major differences in the recovered height maps and the ground-truth surfaces are near the nose, the cheeks and the mouth area.

To provide a more detailed analysis of our method, a more exhaustive set of tests was carried out on fifty images of faces from the database. The absolute height difference comparison plot[6] is shown in Figure 3 (right). The solid line shows the differences calculated on the surfaces integrated after using our method, while the dotted line shows the case of the un-altered gradients. It is obvious that after gradient re-direction the difference among the integrated surfaces decreases considerably, around 53% on average.

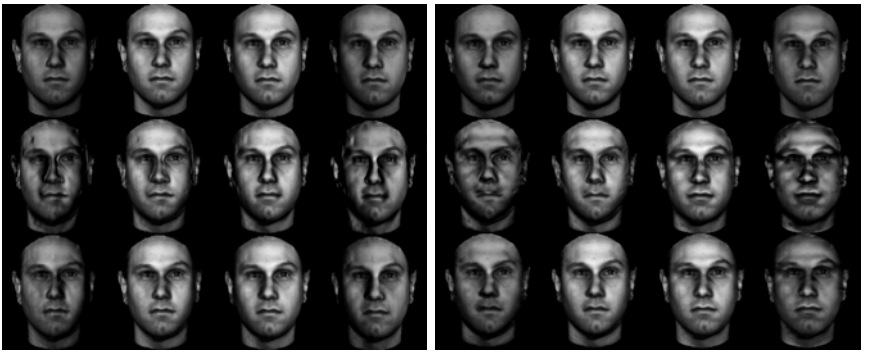


Fig. 4. Comparison for re-illumination tests (see text)

The following set of tests show how our proposed method for re-directing surface normals of faces can be used to produce differently illuminated versions of the face [7]. In Figure 4 we show some results of the re-illumination process. For both sets of images, the top row represents the re-illumination tests using the ground-truth normals, the middle row shows those obtained using the unaltered gradient field and the bottom row those obtained using the re-directed surface normals. From left to right, the light source direction is makes an angle of -45, -25, 25 and $+45$ degrees to the image normal in the horizontal (x) direction for the left-most set of images. In the right-most set of images, the light source is moved in the same manner in the vertical direction. It is interesting to note the similarities between the ground-truth and re-directed gradient re-illuminations. Although the recovered surface does not accurately represent the shape of the image from which it was acquired, the overall shape is good enough to create accurate re-illuminations provided that the light source is not moved by more that 45^o. This results are best when the light source direction is moved in the

[6] For these experiments, the ground-truth surface was generated by integrating the known ground-truth gradient from each image, using the Frankot and Chellappa method. This was done so that all the surfaces were generated on the same basis for comparison purposes. This reduces the biases involved in the integration method.

[7] The lambertian illumination model was applied, assuming as albedo maps the input images.

horizontal direction, and this could be explained as a consequence of the vertical symmetry of human faces. On the other hand, the re-illumination results for the un-modified gradient fields show artifacts of implosion in the area around the nose and mouth. This becomes more severe when the light source moves further away from the viewer direction.

5 Conclusions

We have proposed an algorithm for correcting a gradient field of a face. The aims in doing this are twofold. First, we wish to generate a height map with a global maximum located at a critical point located at the tip of the nose. Second, we aim for force the recovered surface to be convex in accordance with evidence provided by local shape indicators. We have proved that the simple idea of modifying the surface normal directions so as to restore the convexity of imploded features using the constraints derived from the location of a point of global maximum height seems to work well with the recovery of face surfaces. After integration, the recovered shape seems to preserve most salient features, including the nose lips and eye-sockets. The constrain could be used for more general surfaces in a local manner for surface height recovery where there are local regions of implosion.

References

1. Atick, J., Griffin, P. and Redlich, N. (1996), Statistical Approach to Shape from Shading: Reconstruction of Three-Dimensional Face Surfaces from Single Two-Dimensional Images, *Neural Computation*, Vol. 8, pp. 1321-1340.
2. Besl, P.J. and Jain, R.C. (1986) Invariant surface characteristics for 3-d object recognition in range images, *Comput. Vision Graphics Image Proc.*, Vol. 33,pp. 33 - 80.
3. Dovgard, R. and Basri, R. (2004), Statistical symmetric shape from shading for 3D structure recovery of faces, *European Conf. on Computer Vision (ECCV 04)*, Prague, May 2004.
4. Forsythe, D. and Ponce, J. (2001), Computer Vision: a Modern Approach, Prentice-Hall.
5. Frankot, R.T. and Chellapa, R. (1988), A Method for Enforcing Integrability in Shape from Shading Algorithms, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, pp. 438 - 451.
6. Horn, B.K.P. and Brooks, M.J., (1989), Shape from Shading, *MIT Press, Cambridge, MA*.
7. Koenderink, J.J., and Van Doorn, A.J. (1992), Surface Shape and Curvature Scales, *Image en Vision Computing*, Vol. 10,pp. 557-565.
8. Prados, E. and Faugeras, O. (2004), A rigorous and realistic Shape From Shading method and some of its applications, *INRIA Research Report, RR-5133*, March 2004.
9. Worthington, P. L. and Hancock, E. R. (1999), New Constraints on Data-closeness and Needle Map Consistency for Shape-from-shading, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 12, pp. 1250-1267.

The Virtual Point Light Source Model the Practical Realisation of Photometric Stereo for Dynamic Surface Inspection

Lyndon Smith and Melvyn Smith

Machine Vision Laboratory,
Faculty of Computing, Engineering and Mathematical Sciences (CEMS),
University of the West of England, Bristol, BS16 1QY, UK
Telephone 0117 3283578
Lyndon.Smith@uwe.ac.uk
http://www.uwe.ac.uk/cems/research/melsmith/

Abstract. The implications of using commercially available non-collimated and distributed illuminates for the application of photometric stereo to dynamic surface inspection tasks are considered. A new form of lighting model, termed the virtual point light source model, is proposed for modelling real distributed illuminates in relative close proximity. The new technique has application for the two- and three-dimensional inspection of moving surfaces using an innovative technique known as dynamic photometric stereo. Such surface inspection tasks have previously been considered difficult or impossible to undertake using conventional imaging techniques. Experimental results are presented in the paper.

1 Introduction

Previous work [1, 2] has shown how photometric stereo (PS) has useful application for the capture and analysis of fine surface texture and specifically a potential for the inspection of complex surfaces possessing concomitant two- and three-dimensional surface features. The technique enables precisely registered pixel based 2D albedo patterns and 3D surface topographic features and textures to be isolated and subsequently separately analysed [3]. It is important to appreciate that such complex surfaces had previously been considered difficult or impossible to reliably inspect using conventional 2D-image capture and processing techniques. Not surprisingly it has been realised that this approach may have potential application across a range of difficult surface inspection and analysis tasks, including the industrial inspection of textiles, leather, ceramics and wood products [1], as well as new applications in forensic science and medicine [4] using portable hand-held devices. In all cases the richer data set afforded by the utilisation of the PS method is recognised to potentially offer significant theoretical advantage over conventional 2D image intensity or scalar based imaging techniques [1-3]. Before considering issues relating to the practical application of PS it is useful to briefly review the basic theory of conventional 'static' PS. A detailed explanation will not be given here; instead the interested reader is

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 495 – 502, 2005.

referred to [1]. However, given the importance of achieving practical application, particular consideration will be given to those idealised assumptions commonly made in PS literature.

2 Theory of Conventional Static Photometric Stereo

2.1 Two Light Source Photometric Stereo for Planar Surface Inspection

Figure 1 shows two distant point light sources forming a right-angled triangle with an unknown surface normal. The surface is assumed to be Lambertain and the viewing position is held to be distant. Assuming the image intensity under each illuminate to be I_1 and I_2 respectively, then within the plane of the triangle:

$$\text{Albedo} = (I_1^2 + I_2^2)^{0.5} \tag{1}$$

And, $$\text{Topography (i.e. normal sense within the plane)} = \text{Tan}^{-1}\,(I_1 / I_2) \tag{2}$$

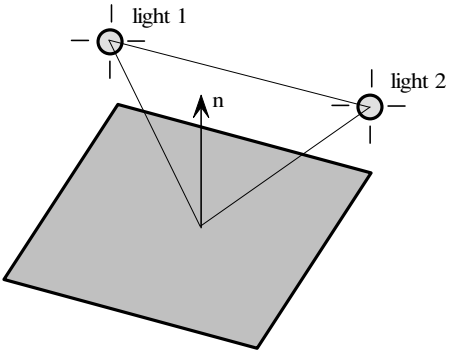


Fig. 1. Two-light photometric stereo schematic

Hence, the method is able to isolate 3D topography and 2D albedo patterns for nominally planar surfaces, commonly encountered in numerous industrial inspection tasks. (Note that in order fully to define non-planar three-dimensional surfaces a third light/image must be introduced.)

3 Dynamic Photometric Stereo

A new form of dynamic photometric stereo (DPS) has recently been proposed [5] in which the conventional static form is extended to allow application to the inspection of fast moving surfaces, typically encountered in industrial quality control tasks. In order to realise a dynamic application it is necessary to either use a single composite image or to acquire multiple images simultaneously. DPS uses different forms of multiplexing in order to allow the simultaneous acquisition of multiple views. Channel isolation (where the term 'channel' is used here to refer to a separate lighting

image combination) may be achieved using methods of spectral, spatial or temporal multiplexing. A detailed description of the DPS technique can be found in [5, 6]. Figure 2 shows a schematic of the lighting and camera configuration used for DPS.

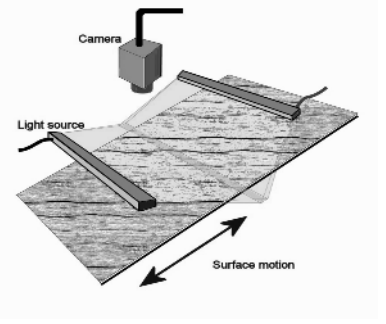


Fig. 2. Web inspection using nearby linear illumination and line scan camera

4 Theoretical Assumptions About Lighting in Photometric Stereo Method

Several factors are fundamental to successful lighting for machine vision. These include angle of illumination, intensity, uniformity, distribution and spectrum of light. Although these factors may be altered and arranged to achieve lighting for highly application specific situations, in all cases the availability of commercial lighting solutions will set a limit on what may be achieved within practical design constraints. As with PS the theory of DPS is based on a number of idealised assumptions. In particular it is assumed that the light sources used are collimated in nature, i.e. take the form of distant point sources. While this may be reasonably well approximated in the laboratory setting, achieving collimated illumination with sufficient uniformity in intensity, over a suitable area for the practical inspection of large fast moving surface, becomes more problematic.

4.1 Real Illuminates

Real illuminates have a finite size and proximity. In order to achieve the necessary intensity distribution required for the imaging of moving surfaces at production rates often exceeding 200m/minute it is often necessary to employ distributed sources in relative close proximity, such as the line lights depicted in Figure 2. Such lights may take the form of florescent tubes, linear arrays of fibre optics or high intensity LED's and would appear to be far from 'point source' in nature.

Modelling real illuminates. First consider the problem associated with the nearby point source, as illustrated by Figure 3. As a result of non-collimation the local position of the source appears to vary between surface locations (i.e. a <> b). Although a calibration may be applied to accommodate this variation, the distributed nature of a real source would seem to suggest that it would be unrealistic to treat it as a point source.

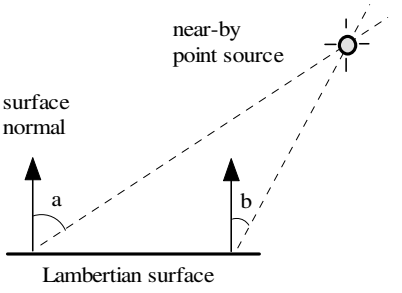


Fig. 3. A near-by point source, where a > b

However, in the case of diffuse reflection, we show here that a nearby-distributed illuminate can in practice always usefully be approximated by an equivalent virtual point source.

The virtual point light source

Consider a Lambertian surface illuminated by a continuous linear illuminate source, rather like a long strip light, as shown by Figure 4, for which it is assumed that the length of the source (L) is significantly greater than the length of the surface (S).

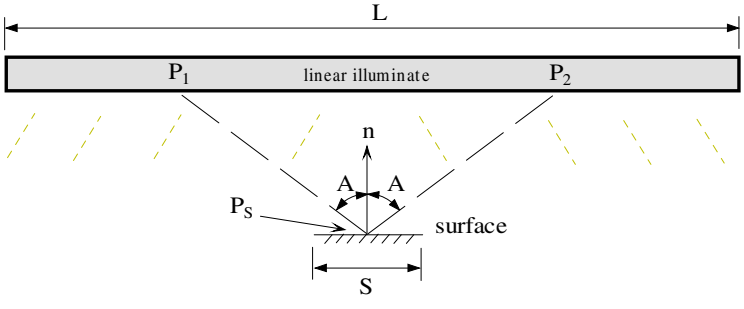
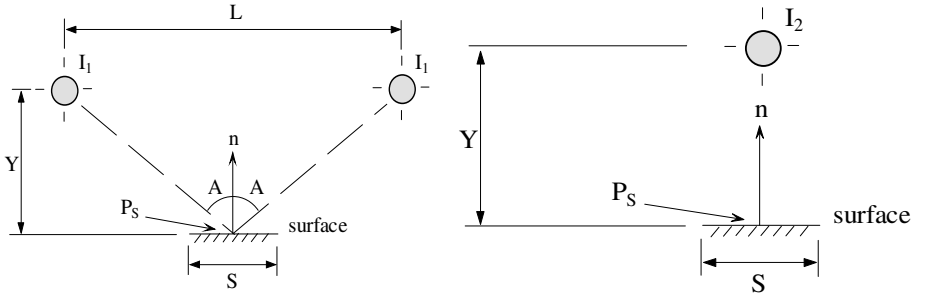


Fig. 4. Distributed linear illumination, where angle A is in Radians

This is not an unreasonable assumption in practice. Any two points, say P_1 and P_2, on the source that are equally spaced either side of a given surface point of interest, P_S, will be at an equal distance and appear with equal intensity, I_1. For $0 < A < \pi/2$ we may consider the strip source to be composed of a large number of such equally spaced sources as $L \rightarrow \infty$. Now, by analogy, consider two point sources of intensity, I_1, as shown by Figure 5(a), for which A is known where we wish to find the equivalent single virtual point source P_V, located above P_S and with intensity I_2, as shown by Figure 5(b).

From Figure 5(a), for a rotation of the surface θ, where to avoid occlusion $\theta < ((\pi/2)\text{-}A)$ the intensity (i) at Ps is given by



(a) Two point sources (b) Equivalent virtual point source

Fig. 5. Virtual point light source

$$i = I_1 Cos(A+\theta) + I_1 Cos(A-\theta) \quad (3)$$

$$i = I_1 [Cos(A+\theta) + Cos(A-\theta)] \quad (4)$$

$$i = I_1 [2CosA.Cos\theta] \quad (5)$$

$$i = I_2 Cos\theta \quad (6)$$

$$\text{Where } I_2 = 2I_1 CosA \quad (7)$$

This result is interesting in that, ignoring occlusion and inverse square effects, i.e. for distant sources, any group of real point sources may be replaced by an equivalent virtual source, as depicted by Figure 6. Furthermore, in the case of a real distributed source, we may consider the source to be the superposition of a large number of symmetrical point sources and as such modelled by a single virtual distant point source. This assumes that the entire illuminate remains visible and is not occluded by surface features or subject to end effects. The situation remains much the same for an extended source, such as a strip light typically used in web inspection, while in close proximity, as shown by Figure 2. For a long (in relation to the illuminated surface) strip light, and a nominally planar surface, in which the amplitude of topographic features is relatively small, these assumptions hold, even if the source is nearby. This is essentially because end-effects may be ignored and by making assumptions about

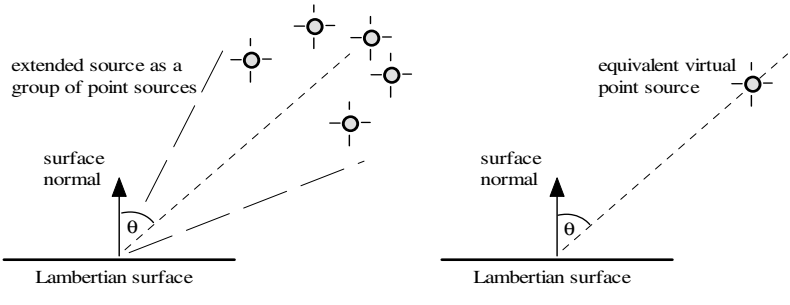


Fig. 6. Approximating an extended source

symmetry it is possible to ignore inverse square effects, as there will always exist symmetrically spaced illuminate points of equal intensity. Even for a distributed source for which L α s, at L > 5Y the variation due to inverse square is found to be less than 2%.

4.2 Experimental Validation of the Virtual Point Light Source

Figure 7 shows a plot of reflected intensity using an extended illuminate in the form of a 122cm long strip light located at a distance of 42cm from a matt surface. Figure 7(a) shows that the intensity function, where θ is the angle between the normal and the light source vector, appears to obey Lambert's law and this is confirmed by Figure 7(b), where a linear regression analysis gives a squared correlation coefficient of better than 99.9% There are important practical implications of these findings in terms of application to dynamic PS. In practice LED line lights are often used in web inspection as they offer good stability and long life. Such illuminates are composed of many small sources, each emitting a cone of light, as depicted by the schematic Figure 8(a).

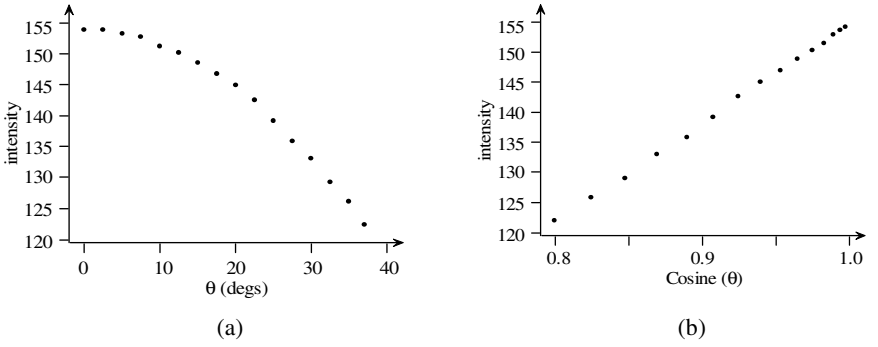
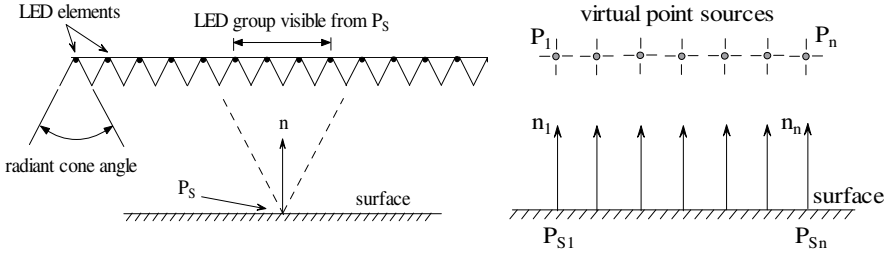


Fig. 7. Matt surface illuminated by an extended source



(a) Schematic of real linear array of LEDS (motion of surface is out of plane)

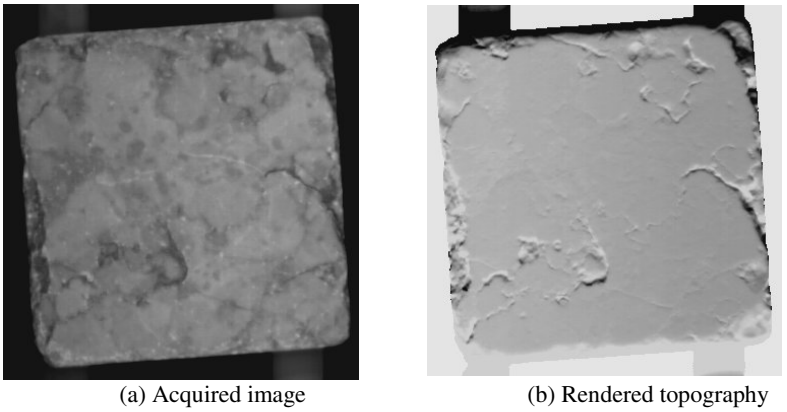
(b) Equivalent virtual point sources

Fig. 8. Modelling a distributed linear source

At any given surface location although a differing group of LEDs will be visible, the size of the group will be similar and will be dependent upon the LED element size, spacing and radiant cone angle. As has been shown, in each case the symmetrical group of LEDS may be replaced by a single virtual source immediately above the surface point of interest, as depicted by Figure 8(b). Hence, it is possible to simulate the nearby-distributed source in terms of known idealised virtual point sources.

5 Dynamic Application Using Line Scan Imaging

Figure 9 shows the results of applying dynamic PS using IR LED line lights, the virtual point light source model and line scan camera acquisition. The ceramic surface was travelling at 30m/minute.



(a) Acquired image (b) Rendered topography

Fig. 9. Dynamic photometric stereo operating at 30m/minute

As can be seen, the albedo is generally very well removed in Figure 9(b) and the detail of the surface topography is clearly visible.

6 Conclusions

The implications of using real illuminates have been considered in the context of the practical application of photometric stereo (PS) to moving surfaces. A new form of virtual lighting model has been proposed, which is able to model real distributed illuminates in close proximity. Experimental results have validated the theoretical findings. The employment of dynamic photometric stereo for the capture and analysis of two- and three-dimensional surface data represents a useful method of enabling surface quality control for rapidly moving production line situations. The technique also offers new potential for the use of free-roaming hand-held devices for surface texture acquisition in medical, forensic science and other evolving fields.

References

1. Smith, M. L., *Surface Inspection Techniques - Using the integration of innovative machine vision and graphical modelling techniques*, Professional Engineering Publishing, ISBN 1-86058-292-3, (2000).
2. Smith, M. L., Smith, L. N., publication agreed for 2004, '*Polished Stone Surface Inspection using machine vision'*, in P. S. Midha (ed), *Complementary Technologies for Stone Processing*, OSNET, (2004).
3. Smith, M. L., Farooq, A. R., Smith, L. N., Midha, P. S., Surface texture inspect tion using conventional techniques applied to a photometrically acquired bump map, Sensor Review, Vol. 20, No. 4, (2000).
4. Smithwick, Q. Y. L., Seibel, E. J.,Depth enhancement using a scanning fiber optical endoscope. *Optical Biopsy IV*, Ed. Robert R. Alfano, Proc. of SPIE Vol. 4613, 222-233, (2002).
5. Smith, M. L., Smith, L. N., Dynamic photometric stereo, submitted to 13th International conference on image analysis and processing, Cagliari, Italy, September 6-8, (2005).
6. Smith M. L., and Smith L. N., (inventors), 'Infra-red photometric stereo', patent application WO03/012412A2, (filed July 2001).

Kernel Spectral Correspondence Matching Using Label Consistency Constraints

Hongfang Wang and Edwin R. Hancock

Dept. of Computer Science, University of York,
Heslington, York, YO10 5DD, UK
{hongfang,erh}@cs.york.ac.uk

Abstract. This paper investigates a kernel spectral approach to the problem of point pattern matching. Our first contribution is to show how kernel principal components analysis can be effectively used for solving the point correspondence matching problem when the point-sets are subject to structural errors, i.e. they are of different size. Our second contribution is to show how label consistency constraints can be incorporated into the construction of the Gram matrices for solving the articulated point pattern matching problem. We compare our algorithm with earlier point matching approaches and provide experiments on both synthetic data and real world data.

1 Introduction

The problem of point pattern matching is to find one-to-one correspondences between two given point-sets and serves as an important component of many computer vision tasks. Graph spectral methods [4] have been used extensively for locating correspondences between feature point-sets, e.g. [13,14,10]. Scott and Longuet-Higgins [13] used a Gaussian weighting function to build an inter-image proximity matrix between feature points and used singular value decomposition (SVD) to locate correspondences. This method fails when the rotation or scaling between the two images being matched is too large. To overcome this problem, Pilu [10] introduces a *feature similarity measure* into the algorithm by incorporating a neighbourhood correlation measure into the proximity matrix. Shapiro and Brady [14] extend the Scott and Longuet-Higgins method and show how correspondences can be located using the eigenvectors of intra-image proximity matrices. Carcassoni and Hancock improve the robustness of the Shapiro and Brady method by using robust error kernels instead of the Gaussian weighting function [1], and exploit image structure using spectral clusters [2].

The location of correspondences between feature points belonging to non-rigid objects is a not only more challenging, but also a potentially more important task. Many existing approaches rely on such information. Examples include the point distribution model (PDM) of Cootes and Taylor [5], and the factorisation method of Tomasi and Kanade [15]. In the literature, many attempts have been described to recover accurate correspondences for non-rigid motion. For example, in [8] the softassign method is used to compute correspondences in a manner that

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 503–510, 2005.

is robust to outliers. In [3], a thin-plate spline is used to model the non-rigid motion of curves and proves successful for point pattern matching.

An interesting source of information that can be used in non-rigid motion, but has received relatively little attention, is that provided by label consistency constraints. In many types of image, the points can be assigned semantic labels to distinguish their identity. Using this information the consistency of pairwise relations can be tested against a scene constraint model. Hence, correspondences which are inconsistent with the model can be rejected. In this paper, we aim to use label consistency information to construct a weighted kernel matrix, and to use this matrix to deliver more robust and computationally effective matching results. Our first contribution is to show how the point proximity matrix can be incorporated into the support function for relaxation labelling. In this way when the label probabilities are updated, then the strength of the proximity relations is brought to bear on the computation of label support. Our second contribution is to show how the label probabilities can be used to refine the point correspondence process using kernel PCA [12]. In our experiments we compare the performance of our algorithm with a number of previous approaches to point pattern matching. We demonstrate that with an appropriate choice of kernel function, the method delivers encouraging performance. In particular, the results are less sensitive to the problems that limit the performance of previous graph spectral methods.

2 Label Process

In the computer vision literature, one of the most extensively studied approaches to the consistent labeling problem involves the use of a discrete or continuous relaxation technique. In the continuous or probabilistic case, each node is assigned an initial weight or probability distribution. Iteratively, the label probabilities or weights are updated until a consistent distribution is reached. The performance of the labelling depends critically on the compatibility coefficients and the support function used to combine evidence in the iterative process. In [7], a dictionary is used, and in [9] the compatibility coefficients are represented as a vector which is learned offline. Here our compatibility model shares some properties in common with the compatibility vector in [9].

Consider the feature point-sets, $\mathbf{y} = \{\mathbf{y}_j\}_{j=1}^{n}$, $\mathbf{y}_j = (y_{j1}, y_{j2})$, and $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^{m}$, $\mathbf{x}_i = (x_{i1}, x_{i2})$ that result from the motion of an articulated object. Here the former set is treated as the model point-set. We augment the feature vectors with a vector of label probabilities, which represent the likelihood of belonging to possible rigid components. Assume there are L labels (rigid components) in each feature point-set. Each image point $\mathbf{x}_i$ can be assigned a label $\theta_i \in \Omega$, where $\Omega = \{\omega_1, \ldots, \omega_L\}$. Denote by $P(\theta_i = \lambda)$ the probability that node $\mathbf{x}_i$ is labeled as λ, $\lambda \in \Omega$. The vector $\mathbf{p}_i = (P(\theta_i = \omega_1), \ldots, P(\theta_i = \omega_L))^T$ represents the probability of assigning each of the possible labels to the point, with $0 \leq P(\theta_i = \lambda) \leq 1$, and $\sum_{\lambda=1}^{L} P(\theta_i = \lambda) = 1$. The matrix P with the label probability vectors as columns, i.e., $P = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N)^T$, represents the

label probability distribution over the entire point-set. Our ultimate aim is to locate correspondences between the two point-sets on the basis of the above-mentioned information by using spectral graph theory. Our label consistency model is derived from the model feature point-set, and the learned label compatibilities used to assign consistent point labels to the "data" point-set. First a label compatibility matrix $R \in \mathbb{R}^{L\times L}$ is constructed so as to embody knowledge of the number of rigid components, i.e. labels, in each image, together with the semantic constraints that apply between each pair of object-labels. It has elements $R_{ij} = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ come from the same rigid component, and is defined to be -1 otherwise. This definition restricts the nodes to give total positive support to the nodes in the same group (i.e. rigid component) and to contribute a negative support to nodes outside the group. The proximity constraint is also acquired from the model image. Further, we assume that in any two consecutive image frames, the relative position of the rigid components of the object under study will not change dramatically. The label probabilities for the data point-set are updated iteratively commencing from a set of initial values. Updating is effected using neighbourhood support. Let us denote the neighbourhood for the point $\mathbf{x}_i$ and its k closest points by $N_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{ik}\}$. The support from the neighbourhood for the label assignment λ_i to point $\mathbf{x}_i$ is:

$$S_{i,\lambda_i} = \frac{\exp\{\sum_{j\in N_i}\sum_{\lambda_j\in\Omega} P(\theta_j = \lambda_j)R(\lambda_i,\lambda_j)W_{ij}\}}{\sum_{\lambda_i\in\Omega}\exp\{\sum_{k\in N_i}\sum_{\lambda_k\in\Omega} P(\theta_k = \lambda_k)R(\lambda_i,\lambda_k)W_{ik}\}} \tag{1}$$

where $R(\lambda_i, \lambda_j)$ are the elements of the label compatibility matrix R. Here the proximity weights W_{ij} using a Gaussian function, and are used to weight the label-support. The label probabilities are then iteratively updated using the formula:

$$P^{(n+1)}(\theta_i = \lambda) = \frac{P^{(n)}(\theta_i = \lambda) + \mu S^{(n)}_{i,\lambda}}{\sum_{\lambda_i\in\Omega}(P^{(n)}(\theta_i = \lambda) + \mu S^{(n)}_{i,\lambda})} \tag{2}$$

where μ is a constant parameter and n is the iteration index.

3 Kernel Spectral Matching

The problem of point pattern matching is that of establishing one-to-one point correspondences between the two data-sets $\mathbf{y}$ and $\mathbf{x}$ extracted from two different images. Ideally, outliers (i.e. extraneous points due to noise) can be removed from the data-sets during matching. Graph spectral methods solve the point correspondence problem by first constructing a weighted graph representation $G(V, W)$ for each data-set, where V is the node set (the image points) and W is the weighted proximity matrix for the nodes that captures the pairwise spatial relationships between image points. One way of constructing the proximity matrix is to use adjacency relationships. Accordingly $W_{ij} = 1$ if the two points are connected by an edge in the graph, and $W_{ij} = 0$ otherwise. Another popular choice is to use the Gaussian function $W_{ij} = e^{-d^2_{ij}/\sigma}$ (e.g. [14] and [13]). Here d_{ij} is the

Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and σ is a constant. In [14], this is explained as to mapping the original 2-D data to a higher dimensional space to capture the structural arrangement of the feature points. Feature correspondences are then found by using eigendecompositions of the proximity weight matrices. When viewed from the perspective of kernel PCA [12], applying a dissimilarity or similarity function to the original data set is equivalent to the process of using a kernel function to map the data into a higher, possibly infinite, dimensional space. Ideally, this mapping interpolates the data in the new space in a manner that is transformationally invariant. Kernel PCA thus appears to provide us with a theoretical basis for spectral pattern matching. Our aim in this paper is to construct a kernel matrix representation that is further constrained by label consistency information. Our idea is to take advantage of kernel PCA to deliver a more stable and more efficient matching process for articulated point matching.

Kernel PCA [12] can be regarded as a non-linear generalization of the conventional linear PCA method. Conventional PCA provides an orthogonal transformation of the data from a high dimensional space to a low dimensional one, which maximally preserves the variance of the original data. This is done by extracting the first few leading eigenvectors from the data-set covariance matrix, and projecting the data onto these eigenvectors. By constrast, kernel PCA first uses a mapping $\mathcal{T} : \mathbf{x} \mapsto \mathbf{\Phi}(\mathbf{x})$ of the data from the original space into a new feature space $\mathcal{F}$ of higher, possibly infinite, dimension before extracting the principal components. In practice, an explicit mapping $\mathcal{T}$ does not always exist so the mapping is performed implicitly by choosing a suitable kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. The kernel K satisfies Mercer's theorem [16]. To extract the principal components of the mapped data, first a covariance matrix needs to be constructed for the data in the feature space $\mathcal{F}$. Suppose that the image data in the space $\mathcal{F}$ is centred, then the corresponding covariance matrix is: $\overline{\mathbf{C}} = \frac{\mathbf{1}}{\mathbf{m-1}} \sum_{\mathbf{i=1}}^{\mathbf{m}} \mathbf{\Phi}(\mathbf{x_i})\mathbf{\Phi}(\mathbf{x_i})^{\mathbf{T}}$. In [12] Schölkopf, Smola, and Müller show that by solving the eigen-equation $m\lambda\alpha = K\lambda$, the p_{th} feature vector, corresponding to the projection of the p_{th} feature point on the eigenspace, takes the form

$$< v^p, \Phi(\mathbf{x}) >= \frac{1}{\sqrt{\lambda^p}} \sum_{i=1}^{m} \alpha_i^p k(\mathbf{x}_i, \mathbf{x}) = \sqrt{\lambda^p}\alpha_n^p. \tag{3}$$

To generalize the method to non-centered data, the kernel function K becomes [12] $K' = (I - ee^T)K(I - ee^T)$ where $e = M^{-1/2}(1, 1, \ldots, 1)^T$. When more than one rigid component is present in the data, each component must be centered onto its own respective subpart centre of movement. To do this, we first compute the mean position (i.e. subgroup centre) corresponding to each label. For the group with label λ, the mean position is given by $\mu_\lambda = \frac{1}{\sum_i P(\theta_i=\lambda)} \sum_i \Phi(\mathbf{x}_i) P(\theta_i = \lambda)$. The covariance matrix then becomes $\overline{C}_{new} = \frac{1}{m-1} \sum_{i=1}^{m} \tilde{\Phi}(\mathbf{x}_i)\tilde{\Phi}(\mathbf{x}_i)^T$, and:

$$\begin{aligned}
\widetilde{\Phi}(\mathbf{x}_i) \cdot \widetilde{\Phi}(\mathbf{x}_i)^T &= (\Phi(\mathbf{x}_i) - \textstyle\sum_\lambda \mu_\lambda P(\theta_i = \lambda))(\Phi(\mathbf{x}_i) - \sum_\lambda \mu_\lambda P(\theta_i = \lambda))^T \\
&= K(\mathbf{x}_i, \mathbf{x}_i) - \textstyle\sum_{\lambda\in\Omega} \frac{P(\theta_i=\lambda)}{\sum_j P(\theta_j=\lambda)} \sum_j P(\theta_j = \lambda) K(\mathbf{x}_i, \mathbf{x}_j) \\
&- \textstyle\sum_{\lambda\in\Omega} \frac{P(\theta_i=\lambda)}{\sum_k P(\theta_k=\lambda)} \sum_k P(\theta_k = \lambda) K(\mathbf{x}_k, \mathbf{x}_i) \\
&+ \textstyle\sum_{\lambda\in\Omega} \frac{P^2(\theta_i=\lambda)}{\sum_j P(\theta_j=\lambda) \sum_k P(\theta_k=\lambda)} \sum_{j,k} P(\theta_j = \lambda) P(\theta_k = \lambda) K(\mathbf{x}_k, \mathbf{x}_j)
\end{aligned} \tag{4}$$

To perform articulated feature point matching, the Gram matrix can be further re-organized to cope with the relative motion of the components. In particular, we weight the kernel matrix $\widetilde{K}$ using label consistency information for each point-set using the formula:

$$\widetilde{K}_{ij} = \sum_{l=1}^{L} P(\theta_i = l)P(\theta_j = l)K_{ij} \tag{5}$$

The covariance matrix $\overline{C}_{new}$ is then computed using (4) with the above resulting matrix in place of the original matrix K and its eigen-decomposition computed. The mapping of the feature vectors $\tilde{\mathbf{y}}_j$ and $\tilde{\mathbf{x}}_i$ are thus computed by using (3) for the respective model and data point-sets. The next step is to compute an association matrix to measure the similarity of each point pair. Assuming the labels on each feature point are independent of each other, the association of the two feature vectors $\mathbf{y}_i$ and $\mathbf{x}_j$ is computed as follows:

$$M_{ij} = \sum_{\lambda=1}^{L} P(\theta_i = \lambda)P(\theta_j = \lambda) \exp\{-d_{ij}^2/\sigma\}. \tag{6}$$

The correspondences are defined as the most similar node pairs. That is, the correspondence for each node $\mathbf{x}_i$ in $\mathbf{y}$ is the node $\mathbf{y}_j = \max_j M_{ij}$.

The matching process is an iterative one in which at each step new label probabilities are incorporated to improve matching. As an increasing number of correspondences are found, the value of the quantity $S = \sum_i e^{-d^2_{\mathbf{x}_i,\mathbf{y}_j}}$, will increase and approach a maximum value. We thus use S as a stopping criterion for the iteration process. The matching process is summarised as follows:

1. `Initialize` P, `threshold` $= t$;
2. `if` $L > 1$ `learn the label semantics from` $\mathbf{y}$;
3. `Compute the Gaussian association matrix` W `for` $\mathbf{x}$;
4. `Run the labeling process, compute` P^{new};
5. `Use` P^{new} `to compute` $\overline{C}_{new}$ `using` (4) `and` M `using` (6);
6. `Compute` $\mathbf{y}_j = \max_k M_{ik}$ `for each point` $\mathbf{x}_i \in \mathbf{x}$;
7. `diff = S - S_old;`
   `return if (diff` $< t$ `or iteration` $>$ `limit); else update` P;
8. `Go to step 3.`

4 Experimental Results

Experiments are performed with both synthetic and real world data. In the rigid motion case, we also compare the proposed algorithms with the algorithms in [14], [13], and the classical MDS [6]. In both rigid and articulated motions, the experiments focus on the performance of the algorithms when the data are subjected to transformations and uncertainties.

We commence by experimenting on synthetic data. Assume that the point sets are subject to a 2-D affine transformation. Given a point-set $X = \{\mathbf{x}_i\}_{i=1}^{n}$ sampled from a rigid object, a synthetic dataset $X' = sRX+\mathbf{t}$ is generated with a predefined parameter set $\Theta = (s, \theta, t_x, t_y)$ with s a scalar, (t_x, t_y) the translation vector, and θ the rotation angle in R. For single components, the second and third rigid components in the multi-label case, the transformation parameter vector Θ is set to $(0.8, 20°, 10, 15)$, $(0.8, 20°, 10, 15)$, and $(1.2, 30°, 10, 15)$, respectively. For the synthesized single component data-set pair, all algorithms give a 100% correct matching except MDS (with 5% error rate). For articulated motion, the matching process involves a label process to label each feature point $\mathbf{x}_i'$ to the corresponding rigid component it comes from. The initial label probabilities are assigned uniformly and the results are listed in Table 4. The experiments on feature points with Gaussian random position jitter is evaluated by first adding a randomly generated 2-D Gaussian distributed noise matrix $D \sim N(\mu, \Sigma)$ on the data point-set; that is, $\mathbf{x}' = \mathbf{x} + D$. Results are displayed in Figure 1. The experimental results are the averages of 100 runs for each different Σ. In the left of the figure, we compare the results of using different matching methods on single component point-sets. The right of the figure is for articulated case with synthetic data-sets and also real data-sets.

To simulate structural errors firstly we delete l consecutive points to simulate occlusion, and secondly we delete l points in random locations to simulate the effect of segmentation errors for experiments on both synthetic and real data-sets. The matching results are shown in Figure 1. Experiments on real world data-sets include first a sequence of infra-red images of a hand (see, in [2]) which has small geometric deformations and a sequence of the CMU house ([2]) in which the point-sets are of different size and has significant positional jitter. Secondly, a pair of images with two rectangular objects moving away from each other, and images of spectacles with moving arm (data-set 4 and 5, Tabel 4, respectively) are included in the experiments. These results are shown in Tables 1 and 4.

From these experiments, it is clear that the kernel PCA approach gives good results when compared with the approaches of Shapiro & Brady [14], Scott & Longuet-Higgins [13], and the MDS method. Moreover, the kernel method is less sensitive to uncertainties than the alternatives.

Table 1. Matching results (Single component, *Numbers of errors*)

	Hand data				CMU House				
Frames	08/25	09/11	09/25	11/25	01/02	01/03	01/04	01/05	01/06
KPCA,Gaussian	6	4	4	11	2	4	2	2	7
KPCA,Polynomial	5	7	6	12	4	5	3	5	13
MDS	35	5	26	27	5	5	25	25	28
Shapiro&Brady	9	6	8	17	3	5	2	2	9
SLH	4	3	5	10	7	6	3	7	9

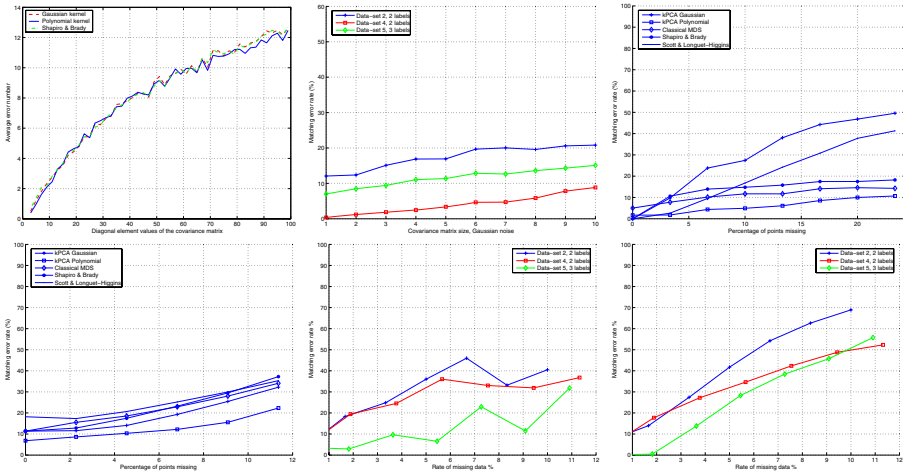


Fig. 1. From top to bottom, left to right: Gaussian jitter, single label synthetic data; Gaussian jitter, multi-label; occlusion, single label synthetic data; occlusion, single label hand sequence; random point missing, multi-label; occlusion, multi-label

Table 2. Matching and labeling results II (Gaussian kernel, *error%*)

Data-set	Num of points	Num of labels	No Label Information	Articulated matching(1)*	Articulated matching(2)**	Labeling
1	10	3	10	0	0	0
2	60	2	53.33	13.33	13.33	0
3	31	2	35.48	0	0	0
4	55	2	18.18	7.27	3.64	1.82
5	53	3	45.28	81.13	3.77	7.55
Note:	*: Results based on label information obtained from the label process; **: Results obtained when correct label information is assumed.					

5 Conclusions

In this paper we have made two contributions. First, we show how the point proximity matrix can be incorporated into the definition of the support function for relaxation labelling and how the label probabilities can be updated with the resulting support. Our second contribution has been to show how label compatibility coefficients can be used to refine the computation of the kernelised proximity matrix for the problems of rigid and articulated point pattern matching. Experimental results reveal that the method offers performance advantages over a number of alternative methods. and gives useful results when there are different moving components in a scene. In the rigid motion case the performance of our algorithm is also comparable to the iterative approaches described in [1,2].

The kernel function used in this paper are possibly not the best choice. One of our future plans to explore kernels that are more stable to structural errors. Our second plan is to do more work on the label process and its interaction with matching. One possibility which has a natural assonance with the kernel method, is to use the heat equation and its spectral solution to model the evolution of label probabilities with time. Work aimed at investigating these points is in-hand and will be reported in due course.

References

1. M. Carcassoni and E. R. Hancock. "Spectral correspondence for point pattern matching". *Pattern Recognition*. 36(2003) pp.193-204
2. M. Carcassoni and E. R. Hancock. "Correspondence matching with modal clusters". *IEEE Tran. PAMI* vol.25 no.12, 2003
3. H. Chui and A. Rangarajan. "A new point matching algorithm for non-rigid registration", *Computer Vision and Image Understanding*, 89:114-141, 2003.
4. Fan R. K. Chung. "Spectral Graph Theory". *Amer. Math. Soc.*, 92, 1997.
5. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. "Training Models of Shape from Sets of Examples". *In Proceedings BMVC*, pp.9-18, 1992.
6. T.F.Cox and M.A.A.Cox. "Multidimensional Scaling". *Chapman and Hall*, 1994.
7. J. Kittler and E. R. Hancock. "Combining Evidence In Probabilistic Relaxation". *Intern. Jour. Patt. Recog. And Arti. Intel.*, vol.3, no.1, pp.29 - 51, 1989.
8. S. Pappu, S. Gold and A. Rangarajan. "A framework for non-rigid matching and correspondence". *Advances in Neural Information Processing Systems 8*, 1996.
9. M. Pelillo and M. Refice. "Learning Compatibility Coefficients for Relaxation Labeling Processes". *IEEE Trans. PAMI*, vol.16, no.9, pp.933 - 945, 1994.
10. M. Pilu. "A direct method for stereo correspondence based on singular value decomposition". *IEEE CVPR*, pp.261-266, 1997.
11. A. Rosenfeld and R. Hummel and S. Zucker. "Scene labeling by relaxation operations". *IEEE Trans. Systems. Man and Cybernetics*, vol.6, pp.420 - 433, 1976.
12. B. Schölkopf, A. J. Smola and K. R. Müller. "Nonlinear component analysis as a kernel eigenvalue problem". *Neural Computation*, vol.10, pp.1299-1319, 1998.
13. G. L. Scott and H. C. Longuet-Higgins. "An Algorithm for Associating the Features of Two Images". *Proc. Royal Soc. London Series B*, vol.244, pp.21 - 26, 1991.
14. L. S. Shapiro and J. M. Brady. "Feature-Based Correspondence - An Eigenvector Approach". *Image and Vision Computing*, vol.10, pp.283-288, 1992
15. C. Tomasi and T. Kanade. "Shape and motion from image streams under orthography – A factorization method". *Tech. Rept TR-92-1270, Cornell University,* 1992.
16. V. N. Vapnik. "Statistical learning theory". *John Wiley & Sons, Inc.,* 1998

Shape Image Retrieval Using Elastic Matching Combined with Snake Model

Chen Sheng and Yang Xin

Institute of Image processing and pattern Recognition,
Shanghai Jiaotong University, Shanghai 200030, P. R. China
chnshn@hotmail.com, yangxin@sjtu.edu.cn

Abstract. Shape-based recovery from image or video databases has become an important information retrieval problem. It is particularly challenging, owning to the difficulty to derive a similarity measurement that closely conforms to the common perception of humans. The goal of the current work is to achieve idea retrieval accuracy with reasonable speed and support for partial and occluded shapes. So, in this paper we introduce the elastic matching that is inspired by Duncan and Ayache combined with snake as a new shape retrieval technique. The elastic matching is to minimize of a quadratic fitting criterion, which consists of a curvature dependent bending energy term and a smoothness term. To reduce the computational complexity, the equation corresponding is only to the minimization of one-dimensional fitting criterion. As a result, the method proposed has the advantage of retrieve resemble objects with reasonable speed and less training samples.

1 Introduction

The problem of similarity-based retrieval of visual shapes has been the subject of much research. Shape retrieval has practical interest for at least two reasons. First, there are many applications in which shapes can be extracted from images with high reliability. Second, there are many sources of visual shapes in addition to images; for example, computer graphics models, CAD models, geographical information systems, MPEG-7 objects, and drawing programs contain shapes or allow shapes to be synthesized from the stored data.

Two essential requirements must be met by a shape retrieval system: accuracy and efficiency. Different approaches make different tradeoffs among accuracy, efficiency, and other desirable characteristics, such as the ability to handle partial or occluded shapes. In this paper, the method is to emphasize accuracy while providing a reasonable level of speed, and also support for partial and occluded shapes.

There are many shape retrieval techniques in the literature: Flickner et al. [1,2] represent shapes by vectors of global feature. A R^*-tree multidimensional access method is used to find database feature vectors that are similar to the query feature vector. This approach is fast, but does not support partial or occluded shapes due to the dependence on global shape properties. Mokhtarian et al. [3] represent each significant segment of the contour by a point in the curvature scale-space representation of the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 511 – 518, 2005.

contour. Matching is performed by aligning these point sets in curvature scale-space. Gdalyahu and Weinshall [4] match shape contours structurally. Each contour is represented as an attributed string corresponding to a sequence of contour segments; the attributed properties are geometric properties of segments. Syntactic matching is performed by computing the minimum edit distance between the strings. Structural shape matching methods are highly effective since they perform a global optimization that takes into account both structural and geometric information. In addition, partial and occluded shapes can be matched. The principal disadvantage of these approaches is that they are computationally expensive. Del Bimbo and Pala [5] integrate indexing with syntactic matching. In their approach, multi-scale representations of the database contours are stored as a graph. Given a query, sequences of segments are matched at the coarsest scale, and if the match is successful, then finer-scale matches are attempted. The method is not orientation invariant since the initial matching is based on segment orientation.

In this paper, a new shape retrieval method is introduced which propose a different elastic matching method. Combined with the snake model, the template is deformed to adjust itself to the shape images. The elastic deformation spent by the template to match the shape images and the matching degree are used to evaluate the similarity between them. It is scaling, rotation and translation invariant and can recover the missing part or remove the occluded part in the shapes. More significant, to reduce the computational complexity, the equation corresponding to the minimization of the fitting criterion has been interpreted as a simple form.

2 The Approach of Shape Representation and Matching

In this approach, suppose we have a one-dimensional shape template:

$$C_{query}(s) = (x(s), y(s))(s \in [0,1]) \tag{1}$$

where: s is the parameter of length along the template. Let Ω be a bounded open subset of R^2, with $\partial\Omega$ its boundary, we have a shape image $u_0 : \overline{\Omega} \to R$. So our purpose is to search for a contour with a shape similar to the original template $C_{query}(s)$.

The template must warp taking into account two opposite requirements. First, it must match the edge of shape in images as closely as possible and the snake model [6] is introduced. It is to minimize the following energy functional:

$$E_{snake}\left(u_0, C_{deformed}\left(s'\right)\right) = E_{int}\left(C_{deformed}\left(s'\right)\right) + E_{ext}\left(u_0, C_{deformed}\left(s'\right)\right) \tag{2}$$

where: $C_{deformed}$ is the deformed template shape with s' the parameter of length, E_{int} is the internal energy that controls the smoothness of the shape and E_{ext} is the external energy that attracts of the template evolving to the edge of object.

Secondly, it needs another energy to measure the deformation of the template also called elastic deformation by us. In this paper, we use a criterion to measure the de-

formation, which consists of a curvature dependent bending energy term and a smoothness term.

The curvature is a key descriptor of the shape in this method because it satisfies the following requirements:

1. The curvature is invariant under rotation and translation.
2. The curvature is a local, scale-dependent feature. A series of shapes can be matched at any desired scale by using a multi-resolution approach.

These allow one to introduce a local bending energy measure of the form:

$$E_{curvature} = \int \left(k_{C_{deformed}}(s') - k_{C_{query}}(s)\right)^2 ds \tag{3}$$

where: $k_{C_{deformed}}(s')$ is the curvature of deformed template $C_{deformed}$ at s' as well as $k_{C_{query}}(s)$.

We also wish the displacement vector field to vary smoothly along the contour [7]:

$$E_{smooth} = \int \left\| \frac{\partial \left(C_{deformed}(s') - C_{query}(s)\right)}{\partial s} \right\| ds \tag{4}$$

where: $\|\cdot\|$ denotes the norm associated to the scalar product $\langle \cdot, \cdot \rangle$ in the space $\Re^2$.

So the criterion is composed of two terms:

$$E_{elastic} = E_{curvature} + \lambda E_{smooth} \tag{5}$$

where λ is a relative weighting factor (a high λ value means heavy smoothing). Actually we have experimented different strategies for weighting the relative balance between smoothness and curvature similarity. One of the successful choices for λ seems to be the heuristically defined adaptive weighting parameter as following [8]:

$$\lambda = \frac{1}{1 + k_{C_{query}}(s)} \tag{6}$$

The idea underlying this parameter definition is to make $E_{curvature}$ preponderant for those contours having characteristic points of high curvature. In the opposite case, the above definition for the parameter λ will make E_{smooth} preponderant.

Duncan [7] finds a displacement field by direct minimization of a discrete form of Equation (5), the resulting displacement vectors in his approach may, however, map points not belonging to the two contours. This problem was solved by Cohen [9]. His mathematical model can be summarized as follow: Given two contours $C_{deformed}$ and C_{query} parameterized by $s' \in [0,1]$ and $s \in [0,1]$, we have to determine a function $f : [0,1] \to [0,1]; s' \to s$ satisfying

$$f(0) = 0 \text{ and } f(1) = 1 \tag{7}$$

And

$$f = arg\ min\, E_{elastic}(f) \tag{8}$$

Cohen obtains the function f, which satisfies Equation (8) and conditions (7), by a variational method. This method finds a local minimum of the function $E_{elastic}(f)$ as the solution of the Eular-Lagrange equation $\nabla E_{elastic}(f) = 0$:

$$\begin{aligned} f''\left\|C_{deformed}'(f)\right\|^2 + k_{C_{query}}\left\langle N_{C_{query}}, C_{deformed}'(f)\right\rangle \\ + \frac{1}{\lambda}\left[k_{C_{query}} - k_{C_{deformed}}(f)\right]k'_{C_{deformed}}(f) = 0 \end{aligned} \tag{9}$$

It is obvious that Equation (9) is complicated and difficult to solve. Our intention is to find a simplified equation, without losing the bending energy and smoothness requirements.

According to Yang [8].

$$\int\left\|\frac{\partial\left(C_{deformed}(s') - C_{query}(s)\right)}{\partial s}\right\|^2 ds \geq \int\left|\frac{\partial\left[f(s) - s\right]}{\partial s}\right|^2 ds \tag{10}$$

Where $f(s) - s$ is the displacement due to the deformation.

Equation (10) establishes that the newly introduced smoothness term:

$$E_{smooth} = \int\left|\frac{\partial\left[f(s) - s\right]}{\partial s}\right|^2 ds \tag{11}$$

Writing the Euler-Lagrange equation for the variational Equation (5) using the smoothness term of Equation (11), leads to a more simple equation:

$$\begin{cases} f'' + \frac{1}{\lambda}\left[k_{C_{query}}(s) - k_{C_{deformed}}(f(s))\right]k'_{C_{deformed}}(f) = 0 \\ \qquad + boundary conditions \end{cases} \tag{12}$$

3 Similarity Measure

After the deformed template reaches convergence over a shape image. We need to measure how much the shape in image is similar to the original template, and it is a fuzzy concept. In order to measure it, the first we need to think about is overlapping between the deformed template and the shape image. The second need to be thought about is the elastic deformation between the original template and the deformed template. An example was shown in Fig.1.

The similarity measurement in this paper is scaling, rotation, translation invariant. In Fig.2, the five-tip star template is made warp over a set of rotated star images that are also scaled up or down. From Table.1, it can be noticed that the energy of similarity measurement are fairly equal.

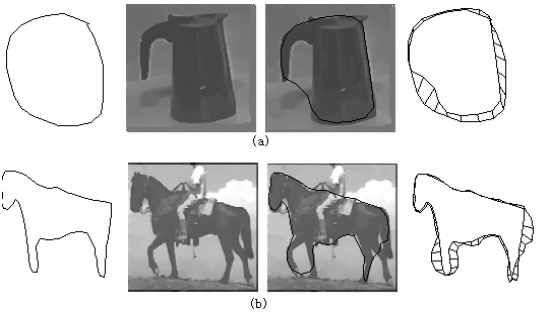


Fig. 1. Elastic matching and deformation

Fig. 2. Template of a five-tips star and images rotated and scaled up or down

Table 1. Energy of the star template to match the shape images in Fig.2

Star	Energy (elastic matching)
(a) original	0.67
(b) rotated by 20°	0.62
(c) rotated by 30°	0.52
(d) rotated by 40°	0.63
(e) scaled down	0.61
(f) scaled up	0.68
(g) rotated and scaled	0.70

4 Results

The proposed method was implemented under the Visual C++ 6.0 system on a P4-1.4GHz PC. A shape database as shown in Fig.3 including nine categories with 11 shapes in each category was used to test the proposed method. In Fig.4, for each template, the 11 most similar shapes were retrieved ranked by the similarity measurement. It is clear that the approach is tolerant to occlusion although there are some missing or occluded parts in some categories such as humans, rabbits and

hands shapes. The retrieval time was well under 3s for each template to rank the 99 shapes. Fig.5 compared the % retrieval accuracy of the proposed method with other methods such as: Curvature scale space descriptors (CSSD), Centroid-contour distance (CCD). Precision is defined as the ratio of the number of retrieved relevant shapes to the total number of retrieved shapes. Recall is defined as the ratio of the number of retrieved shapes to the total number of relevant shapes in the whole database.

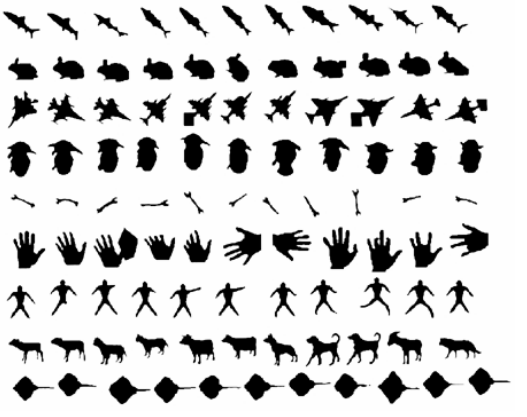


Fig. 3. Shape image database for test

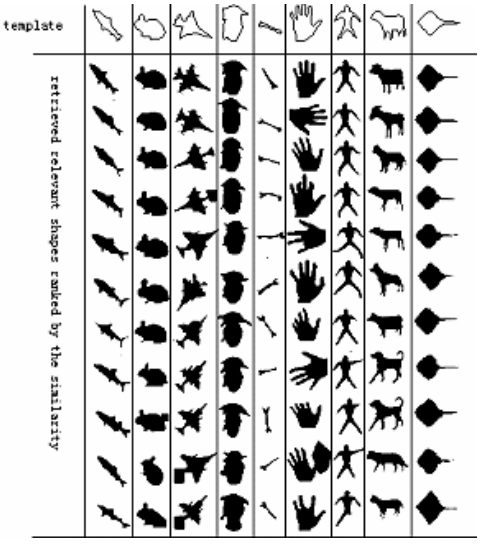


Fig. 4. Retrieval results for the shape database

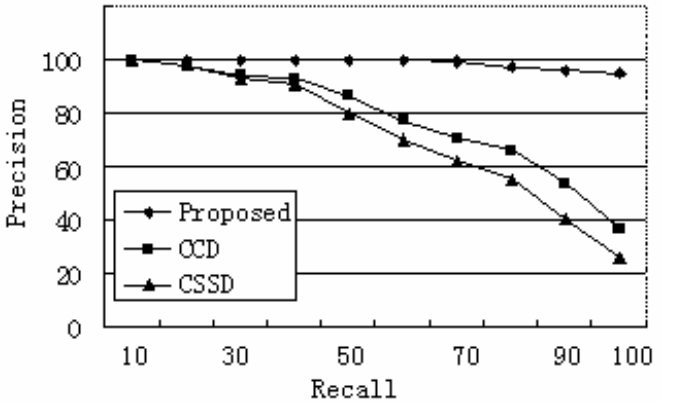


Fig. 5. Average retrieval accuracy

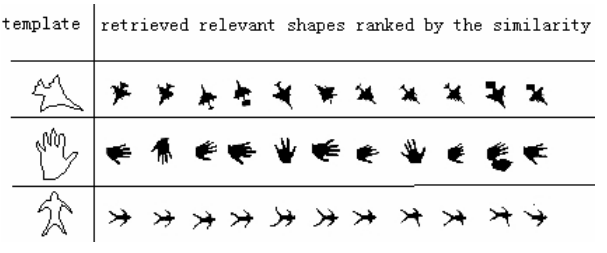


Fig. 6. Retrieval results for the shape database

More significantly, in order to demonstrate the method is scaling and rotational invariance, the experiment was repeated with the images that are scaled down to 50% and rotated by 90° . From Fig.6, we can see that the result is identical regardless of their scaling and rotation.

5 Conclusions

A new elastic matching method for shape image retrieval has been presented in this paper. Since the similarity measurement is invariant to translation, rotation and scale variations, the proposed method can handle the invariance requirement. In addition, it allows one to handle situations in which part of the shape information is missing or occluded. Finally, since no start point problem when matching shapes, the computational efficiency is not degraded. It has been proven by various experiments.

Acknowledgments

This work was partially supported by National Science Research Program of China (No. 2004BA714802) and Shanghai Science and Technology Development Foundation (034119820).

References

1. C. Faloutsos, R. Barber,M. Flickner, J.: Efficient and effective querying by image content. J. Intell. Inform. Systems **3** (1994) 231–262
2. M. Flickner et al.: Query by image video content: The QBIC system. IEEE Comput. (1995) 28 23–32
3. S. Abbasi, F. Mokhtarian.: Enhancing CSS-based shape retrieval for objects with shallow concavities. *Image Vision Comput.* (2000) 18 199–211
4. Y.Gdalyahu and D.Weinshall.: Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Trans. Pattern Anal. Mach. Intell.* (1999) 21 1312–1328
5. A. Del Bimbo and P. Pala.: Shape indexing by multi-scale representation. *Image Vision Comput.* (1999) 17 245–261
6. Kass M, Witkin A and Terzopoulos D.: Snakes: active contour models. International Journal of Computer Vision (1987) 1 321-331
7. J.S. Duncan, R. Owen, P. Anandan.: Shape-based tracking of left ventricular wall motion. Computers in Cardiology 1990, IEEE Computer Society, Chicago, Illinois, September 1990 23-26
8. Yang Xin, Bart Truyen. Hierarchical contour matching in medical images. Image and Vision Computing (1996) 14 417-433
9. Cohen I, Ayache N, Sulger P.: Tracking points on deformable curves. Proc Second Euro. Conf. Computer Vision 1992 Santa Margherita Ligure, Italy, may 1992

Image-Based Relighting of Moving Objects with Specular Reflection

Hanhoon Park[1], Jong-Il Park[1], and Sang Hwa Lee[2]

[1] Division of Electrical and Computer Engineering, Hanyang University, Seoul, Korea
{hanuni, jipark}@mr.hanyang.ac.kr
[2] Department of Electrical Engineering, Seoul National University, Seoul, Korea
lsh@ipl.snu.ac.kr

Abstract. In the fields of Augmented Reality (AR) and Virtual Reality (VR), inserting an object into a scene (real or virtual) requires proper matching of their lighting conditions. If not, the resulting image may look unnatural. In particular, it is important to describe the position and shape of specular reflection accurately if the object has specular reflection. In this paper, we propose an approach to relighting a moving object based on the separation of specular and diffuse reflection. To relight an object, two or more images taken under the condition that the position of the object is fixed but the lighting condition is different, we call *synchronized images*, are required. However, it is impossible to obtain such images in case of a moving object. Therefore, we propose a method that computationally obtains the synchronized images using the consecutive fields of a controlled video sequence containing a moving object. For example, if the virtual $(n+1)-th$ field is interpolated from $n-th$ field and $(n+2)-th$ field using the motion compensation technique, both the virtual $(n+1)-th$ field and the real $(n+1)-th$ field have the condition that the position of the object is fixed. If the virtual and real image have different lighting condition, the method applied to static object is applicable to moving object as it is. After the specular and diffuse reflection are separted, the relit image is synthesized using the linear interpolation and morphing technique. The experimental results of applying the proposed method to real and synthetic images are given. We verify the effectiveness of the proposed method by comparing the resulting image with a ground-truth image.

1 Introduction

Matching the lighting condition of the involved images is a necessary step in synthesizing photorealistic images from different image sources. In particular, it is important but hard to handle the specular reflection, which is necessary to render an object realistically. For convenience, most of computer graphics algorithms have assumed that the object has no specular reflection (i.e. the surface is Lambertian) [1,2,3,4]. Obviously, this assumption does not fully satisfy the reality requirements in computer graphics. Recently, several approaches associated with image based relighting (IBL) have been reported to handle the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 519–526, 2005.

specular reflection independently by separating specular reflection from diffuse reflection. Many approaches for separating specular and diffuse reflection have been proposed [5,6,7,8,14] and successfully applied to relighting static objects [9].

There have been several approaches for changing the lighting condition of an image [11,12,13]. They require capturing numerous images using a number of light sources and focus on relighting static objects. Recently, an efficient approach for relighting a static object in which only two images are required was proposed [9]. To relight a static object, two or more images that are taken in the condition that the position of the object is fixed but the lighting condition is different are required [9]. By comparing the values of the corresponding pixels on the images with each other, specular and diffuse reflection are separated and used for relighting the object independently.

This paper is an extension of [9] toward coping with a moving object. Notice that the algorithm for relighting a static object can be used for relighting a moving object as it is if we can obtain a pair of synchronized images with different lighting condition from a video sequence containing a moving object, Therefore, we propose a method that computes a pair of synchronized images from the consecutive fields of a controlled video sequence of a moving object, where the lighting conditions of even/odd field are alternating. The synchronized images are obtained by applying a motion-compensated interpolation algorithm to the controlled video sequence.

2 Image-Based Relighting Based on the Separation of Specular and Diffuse Reflection

In this section, we briefly explain the process of relighting a static object based on the separation of specular and diffuse reflection. Given two images on which a static object is taken in the different lighting condition, the specular and diffuse reflection are separated. To relight the static object, a linear interpolation algorithm and a morphing algorithm are applied to the resulting specular and diffuse reflection, separately. The shaded region in Figure 4 depicts the overall process for relighing a static object.

2.1 Separation of Specular and Diffuse Reflection

Specular reflection is an important factor for describing an object realistically but a troublesome in the field of computer vision. Most of computer vision algorithms have assumed that the object has no specular reflection. To handle specular reflection effectively, it is necessary to separate specular reflection from diffuse reflection. In the literature, the algorithms for separating specular and diffuse reflection are mainly divided into two categories: color-based [5,6,14]; polarization-based [7,8]. Both kinds of algorithms have shown satisfactory results but the former requires lots of images for processing, the latter has a difficulty in obtaining suitably polarized images. In this paper, the more effective method are used, which is a color-based method but requires only a few images [9]. The

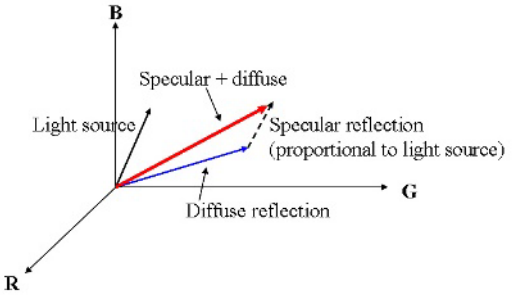


Fig. 1. Property of specular and diffuse reflection

method is simpler and more effective than the method proposed by Lin and Shum [6] which is based on complicated spectrum analysis.

Two images are used on which an object is taken in the different lighting condition. Given *image A* and *B*, the pixel values of each image are denoted by the sum of specluar and diffuse reflection [14]. Then, the specular and diffuse reflection can be represented in RGB color space as shown in Figure 1 because the intensity of specular reflection is proportional to the intensity of light source [10]. If the intensity of an arbitrary pixel on *image B* is lower than that of the corresponding pixel on *image A*, it is regarded that the pixel has no specular reflection while the correspoding pixel may have specular reflection. Given a light source as $\boldsymbol{E_k}(k = r, g, b)$, the specular reflection is represented as $t\boldsymbol{E_k}(t \geq 0)$. the specular reflection can be separated from the diffuse reflection using Eq. (1) which is derived from the fact that the RGB values of diffuse reflection are changed while the ratio among them is preserved when the lighting condition is changed [10].

$$\frac{r(I_A) - t\boldsymbol{E_r}}{r(I_B)} = \frac{g(I_A) - t\boldsymbol{E_g}}{g(I_B)} = \frac{b(I_A) - t\boldsymbol{E_b}}{b(I_B)} \tag{1}$$

where r(*), g(*), b(*): the value of red, green, and blue.

2.2 Synthesis of Virtual Specular and Diffuse Reflection

Diffuse reflection by a virtual-light-source C can be computed by linearly interpolating diffuse reflection by a real-light-source A and diffuse reflection by a real-light-source B. Image morphing is used to synthesize the specular reflection by a virtual-light-source. In particular, voxel-based morphing is used because it can be consistently applied to a whole image and thus easy-to-use compared with polygon-based one which requires to discriminate the object area from another area and approximate it to a polygon.

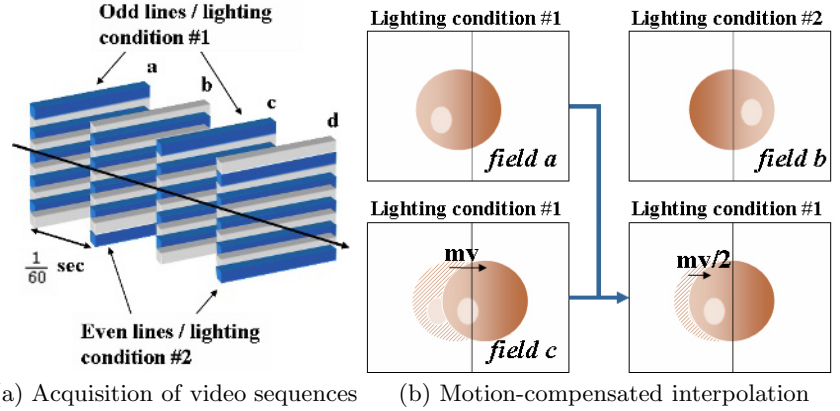
Given two objects which are represented by its center and size at one-dimensional space, the center and size of the intermediate object can be computed by linearly interpolating those of the objects. Two-dimensional image morphing is achieved by applying the one-dimensional morphing to each row

of an image, independently. Specular reflection by a virtual-light-source is computed by applying the two-dimensional image morphing algorithm to specular reflections by two real-light-sources which have the different position from each other. The morphing algorithm does not work if the height of target objects are different from each other. Therefore, after computing the minimum-bounding-box including specular region and normalizing the size of the box to a fixed size, the morphing algorithm is applied to the normalized region. The transformed specular region is located at the virtual image by applying linear interpolation algorithm to the center, width, and height of the specular area by two real-light-sources, respectively [9].

3 Computing Synchronized Images for Moving Object

In this section, we explain how to relight moving objects. The method of relighting a moving object is the same as that for a static object. However, since we cannot obtain a pair of images required for relighting a moving object from a video sequence directly, we compute the required images from a controlled video sequence of a moving object. Different lighting conditions are applied to the odd-field and even-field of an interlaced video sequence alternatively using a controlled strobe light. Then, the required images are synthesized using a temporal interpolation algorithm.

The motion of an object between the fields is assumed to be small because the temporal gap between the fields of video sequences is very small. Thus, if the virtual $(n+1)-th$ field is interpolated from $n-th$ field and $(n+2)-th$ field using the motion compensation technique, the virtual $(n+1)-th$ field and the real $(n+1)-th$ field satisfy the condition that the position and shape of the object is fixed. Because the virtual and real images have different lighting



(a) Acquisition of video sequences (b) Motion-compensated interpolation

Fig. 2. Computation of a pair of images for relighting a moving object using the motion-compensated interpolation.

condition, a pair of images required for relighting a moving object are obtained. Figure 2 depicts the procedure of obtaining the required images.

A simple block matching algorihtm is used to estimate the motion of an object. In Figure 2, an image is created on which an object is located in the same position as the object in *field b* by moving an object from the position of the object in *field a* with a half of the motion vector of the object between *field a* and *c*. More sophisticated algorithms would be necessary for dealing with complex motion and occlusion problems, which would be valuable for future work.

An additional line interpolation algorithm is employed because the image synthesized from *field a* and *c* has different even/odd lines from *field b* as shown in Fig 2. We use a simple vertical averaging filter with the kernel of (0.5, 0, 0.5). If the image includes a lot of high-frequency information, more sophisticated algorithms might be required.

4 Experimental Results

To obtain the video sequences used in the experiments, we captured a moving object in a darkroom. The strobe light was used which was synchronized with double frame-rate of a camera by a external trigger and thus the even and odd fields of video sequences have different lighting condition from each other. Figure 3 shows the experimental equipments used in our experiments. The images are at a resolution of 256 by 256 pixels and true-colored.

We take the procedure of Figure 4. From consecutive even and odd fields of the video sequences, a pair of synchronized images where an object is located at the same position but has different light condition from each other were first interpolated. Next, the specular and diffuse reflection were separated from the images. To relight an object, we finally applied a linear interpolation algorithm to the resulting diffuse reflection and a morphing algorithm to the resulting specular reflection, separately. The process was repeated on the field-by-field basis.

Figure 5 shows the results of computing the intermediate image from the two consecutive synthetic images using the motion-compensated interpolation

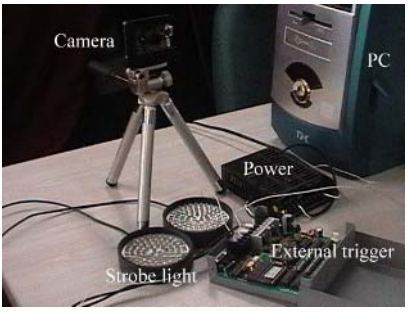


Fig. 3. Experimental equipments

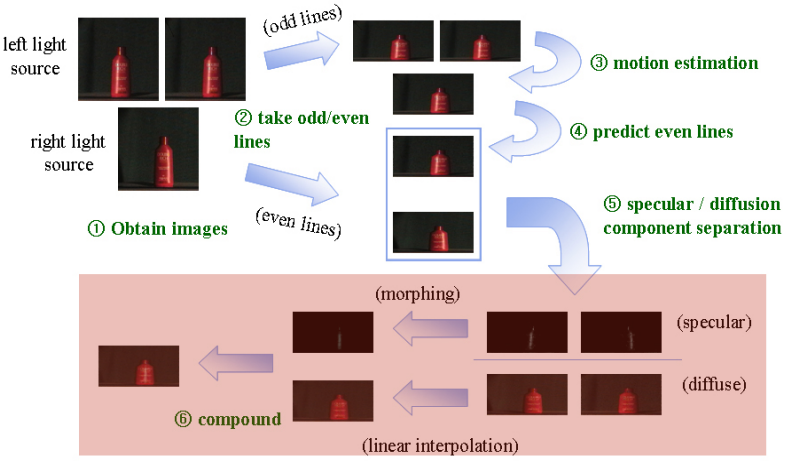


Fig. 4. Procudure for relighting a moving object. The shaded region depicts the process for relighting a static object

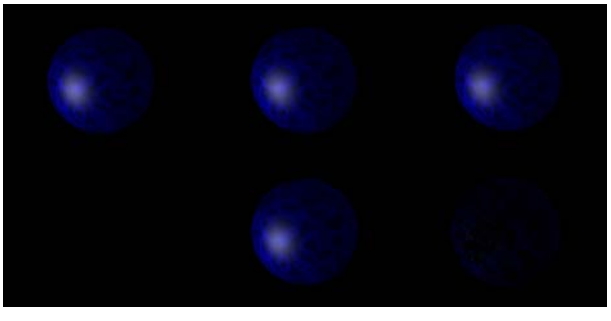


Fig. 5. Motion-compensated interpolation. Left-top: original n-th image. Right-top: original (n+2)-th image. Middle-top: the (n+1)-th computed image. Middle-bottom: ground truth image. Right-bottom: difference between the ground truth image and the (n+1)-th computed image (PSNR = 33.08 dB).

algorithm which are explained in the previous section. The intermediate images were accurately computed (PSNR = 33.08 dB).

Figure 6 shows the results of relighting real and synthetic images based on the proposed relighting algorithm. In case of real images, the relit images were a little bit deteriorated by the effect of brightness saturation, shadow, incomplete specular reflection, and so on. However, the position and shape of synthesized specular reflection are very natural. The overall quality of the synthesized relit image is quite compelling when taking the fact that only the two images was used into the consideration.

To demonstrate the performance of the proposed strategy for a moving object, The result of relighting a moving object was compared with a ground-truth

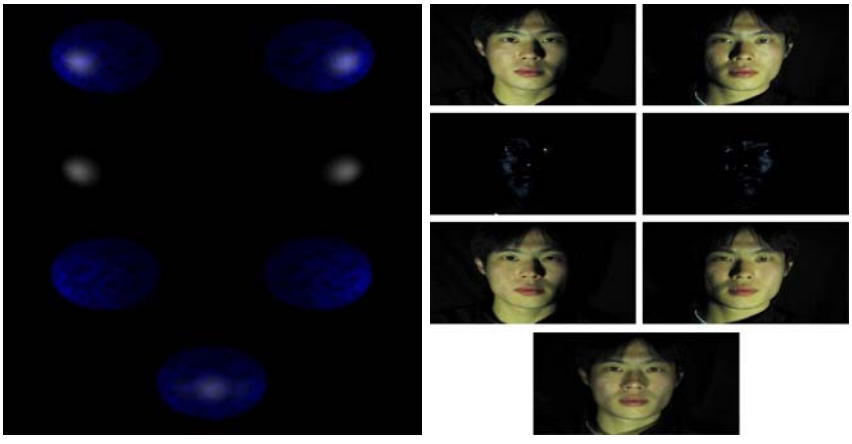


Fig. 6. Results of relighting a synthetic and real image based on the separation of diffuse and specular reflection. 1st-row images: synchronized images. 2nd-row images: the specular images computed from the synchronized images. 3rd-row images: the diffuse images computed from the synchronized images. 4th-row image: the relit images.

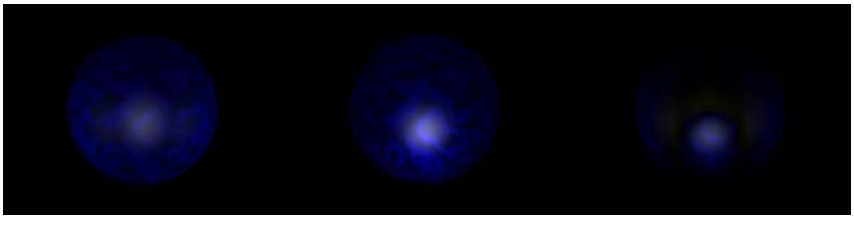


Fig. 7. Comparison of the relit image and a ground-truth image. (a) the result of relighting a moving object, (b) ground-truth image, (c) the difference of the two images (PSNR = 27.87 dB). The specular reflection of the relit image is visually compelling but a little darker than the specular reflection in the ground-truth image. It is assumed that the problem happens because the intensity of specular reflection is different according to its position.

image as shown in Figure 7. The relit image was a little bit blurred compared to a ground-truth image but the difference was quite small.

5 Conclusion

In this paper, we proposed a method of relighting a moving object based on the separation of specular and diffuse reflection. The method is an extension of the method for a static object [9]. Using controlled light sources and interlaced video sequence, we have successfully generated a pair of synchronized images with different lighting condition. Then, the conventional method for a static object was applied. The effectiveness of the proposed method was demonstrated through a variety of experimental results.

The relit image was slightly different from a ground-truth image but visually convincing. In the future, interpolation algorithms suited for the proposed relighting method should be further explored. The method cannot fully cope with the change of shape of specular reflection between consecutive fields. More sophisticated methods towards resolving those problems should be investigated in the future.

Acknowledgement

This study was supported by a grant(02-PJ3-PG6-EV04-0003) of Ministry of Health and Welfare, Republic of Korea.

References

1. Fournier, A., Gunawan, S., Romanzin, C.: Common Illumination between Real and Computer Generated Scenes. Proc. of Graphics Interface (1993) 254-262
2. Gibson, S., Murta, A.: Interactive Rendering with Real-World Illumination. Proc. of Eurographics Workshop on Rendering (2000) 365-376
3. Loscos, C., Drettakis, G., Robert, L.: Interactive Virtual Relighting of Real Scenes. IEEE Transactions on Visualization and Computer Graphics (2000) 289-305
4. Kim, H.: Virtualizing Facial Images Considering Lighting Condition. MS thesis, Hanyang University, Seoul, Korea (2001)
5. Klinker, G.J., Shafer, S.A., Kanade, T.: The Measurement of Highlights in Color Images. International Journal of Computer Vision (1990) 7-32
6. Lin, S., Shum, H.Y.: Separation of Diffuse and Specular Reflection in Color Images. Proc. of CVPR (2001) 341-346
7. Wolff, L.B.: Using Polarization to Separate Reflection Components. Proc. of CVPR (1989) 363-369
8. Nayar, S.K., Fang, X., Boult, T.E.: Removal of Specularities Using Color and Polarization. Proc. of CVPR (1993) 583-590
9. Lee, B.-H.: Image-Based Relighting Using Separation of Specular and Diffuse Reflectance. MS thesis, Hanyang Univesity, Seoul, Korea (2004) (in Korean)
10. Lee, H.-C.: Method for Computing the Scene-Illuminant Chromaticity from Specular Highlights. Journal of the Optical Society of America A (1986) 1694-1699
11. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., Sagar, M.: Acquiring the Reflectance Field of a Human Face. Proc. of SIGGRAPH (2000) 145-156
12. Haeberli, P.: Synthetic Lighting for Photography. (1992) Available on http://www.sgi.com/grafica/synth/index.html
13. Wong, T.-T., Heng, P.-A., Or, S.-H, Ng, W.-Y.: Image-Based Rendering with Controllable Illumination. Proc. of Eurographics Workshop on Rendering (1997) 13-22
14. Shafer, S.A.: Using Color to Separate Reflection Components. Color Research and Applications (1985) 210-218

Modeling of Elastic Articulated Objects and Its Parameters Determination from Image Contours

Hailang Pan, Yuncai Liu, and Lei Shi

Institute of Image Processing and Pattern Recognition,
Shanghai Jiao Tong University, P.R.China, 200030
{panhailang, whomliu, s10030322014}@sjtu.edu.cn

Abstract. This paper presents a new method of elastic articulated objects (human bodies) modeling based on a new conic curve. The model includes 3D object deformable curves which can represent the deformation of human occluding contours. The deformation of human occluding contour can be represented by adjusting only four deformation parameters for each limb. Then, the 3D deformation parameters are determined by corresponding 2D contours from a sequences of stereo images. The algorithm presented in this paper includes deformable conic curve parameters determination and the plane, 3D conic curve lying on, parameter determination.

1 Introduction

Objects in motion analysis are mainly rigid objects, articulated objects, nonrigid objects and so on. In computer vision research, motion analysis has been largely restricted to rigid objects. However, in the real world, motion of nonrigid objects is the rule[1]. In the past decade, there has been a growing interest in the study of nonrigid motion. In nonrigid motion analysis, dynamic shape modeling provides a mechanism for fitting and tracking visual data. Using deformable models, the seemingly unstructured elastic motion can be compactly represented by a small number of parameters. The task of elastic motion recovery is reduced to the problem of the parameters determination.

In the past researches, a number of method for human deformable body modeling have been proposed. Nahas [2] use B-Spline surfaces to model lissom movements of body and face. Petland [3] introduced a model of elastic nonrigid motion. This model is based on the finite element method. Min [4] proposed a layered structure of the human body to provide a realistic deformation scheme for the human body model. Plankers[5]developed a framework for 3D shape and motion recovery of articulated deformable objects. Smooth implicit surfaces, known as metaballs, are attached to an articulated skeleton of the human body and are arranged in an anatomically-based approximation. Sminchisescu [6] built a human body model which consists of kinematic 'skeletons', covered by 'flesh' built from superquadric ellipsoids. A typical model has around 9 deformable shape parameters for each body part. Apuzzo[7] presents simplified Model of a Limb. Only three ellipsoidal metaballs are attached to each limb skeleton and arranged in an anatomically-based approximation. Each limb has twelve deformation parameters.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 527 – 534, 2005.

Elastic articulated objects are the combination of articulated objects and nonrigid objects. Human body is a typical elastic articulated objects. Our research focuses on two major points: 3D human body modeling and its parameters determination from stereo image contours. The proposed human limb model is composed of two layers: a skeleton layer and a body occluding contour layer. The skeleton layer represents the skeletal structure of the human body, which is composed of the sticks and joints which linking these sticks. The body occluding contour layer is expressed by 3D conic curves, attached to an articulated skeleton and set in an contour-based approximation. The body occluding contour layer deforms with the motion of the skeleton. We can concisely deform the body occluding contour layer during animation by adjusting only four deformation parameters for each human limb. The human arm model is equally applicable to other vertebrates, such as horses, cows and so on. There are two aspects of image-based 3D human arm parameters determination. The first is the skeleton motion parameters estimation. The second is the deformable occluding contours deformation parameters determination. We establish the equations of human arm's movement and deformation, analyze the condition for a solution and solve the nonlinear equations.

One motivation of our research is to build a body model that properly describes human body deformation from a small number of parameters and human 3D shape and motion analysis from 2D image sequence.

2 Human Arm Modeling

2.1 Establishment of Conic Curves

This algorithm uses a inducting form to express a 2D conic curve restricted by three straight lines [8](see Fig.1).

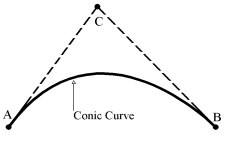


Fig. 1. 2D conic curve

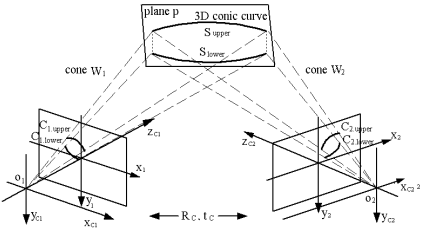


Fig. 2. Coordinate systems and two images of a 3D conic curve

Inducting form of the 2D conic curve is a family of conic curves, which pass through vertices A , B and tangent to the straight lines AC, BC. They have the equation of the form

$$(a_0 + a_1x + a_2y)(b_0 + b_1x + b_2y) = \rho(u_0 + u_1x + u_2y)^2 \tag{1}$$

We can deform the conic curve by adjusting the parameter ρ .The equation (1) can also be represented by the quadratic equation (3).

2.2 Conics-Based Stereo

The geometry of the sensors is shown in Fig.2. In stereo vision, we have two coordinate systems c_1 and c_2 associated with two cameras. The relative position and orientation of c_1 and c_2 are described by a rotation R_c followed by a translation t_c :

$$X_{c2} = R_c X_{c1} + t_c \tag{2}$$

where X_{c1} and X_{c2} are the coordinates of any point in space respectively by c_1 and c_2 frames: $X_{c1}= [x_{c1}, y_{c1}, z_{c1}]^T$ and $X_{c2}=[x_{c2}, y_{c2}, z_{c2}]^T$.

The optical center o_1 (or o_2) is the origin of the camera frame c_1 (or c_2). The z axis, o_1z_{c1}(or o_2z_{c2}), is the optical axis and the image plane, x_1y_1 plane (or x_2y_2 plane), is parallel to the x_{c1} y_{c1} plane (or x_{c2} y_{c2} plane) at a distance f from the origin. We suppose the cameras' intrinsic parameters are known and the pixel coordinates have been transformed to x_1 y_1 (or x_2 y_2) coordinates.

Suppose there is a 3D conic curve S in space. Its two projections on two images are represented by two quadratic form C_1 and C_2 [9] :

$$C_i(x_i, y_i) = X_i^T Q_i X_i = 0 \quad i = 1,2 \tag{3}$$

where $X_i=[x_i,y_i,1]^T$.(x_i,y_i) are the 2D coordinates in two images.Two 2D conic curves, projected from the 3D conic curve S and observed by two cameras,are extracted and represented by the matrixes Q_1 and Q_2 respectively

$$Q_i = \begin{bmatrix} (a_{i1}b_{i1} - \rho_i \cdot u_{i1}^2) & (a_{i1}b_{i2} + a_{i2}b_{i1} - 2\rho_i \cdot u_{i1}u_{i2})/2 & (a_{i0}b_{i1} + a_{i1}b_{i0} - 2\rho_i \cdot u_{i0}u_{i1})/2 \\ (a_{i1}b_{i2} + a_{i2}b_{i1} - 2\rho_i \cdot u_{i1}u_{i2})/2 & (a_{i2}b_{i2} - \rho_i \cdot u_{i2}^2) & (a_{i0}b_{i2} + a_{i2}b_{i0} - 2\rho_i \cdot u_{i0}u_{i2})/2 \\ (a_{i0}b_{i1} + a_{i1}b_{i0} - 2\rho_i \cdot u_{i0}u_{i1})/2 & (a_{i0}b_{i2} + a_{i2}b_{i0} - 2\rho_i \cdot u_{i0}u_{i2})/2 & (a_{i0}b_{i0} - \rho_i \cdot u_{i0}^2) \end{bmatrix} \quad i = 1,2 \tag{4}$$

Let $X_w=[x_w,y_w,z_w,1]^T$. (x_w,y_w,z_w) is a point in the world coordinate system.Then X_i and X_w are related to each other through the following equation:

$$z_{ci}X_i = z_{ci}\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{ci} \\ y_{ci} \\ z_{ci} \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} R_i & t_i \\ 0^T & 1 \end{bmatrix} \cdot \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = M_i X_w \tag{5}$$

Where the matrixes (R_i , t_i), called the extrinsic parameters, is the rotation and translation which relates the world coordinate system to each camera coordinate system. f is the focal length of the cameras.

Substituting the equation (5) into the equation (3), we obtain the representations of the two cones W_1 and W_2 passing through the 2D conic curve C_1(or C_2) and the camera center o_1(or o_2):

$$\begin{cases} W_1(x_w, y_w, z_w) = X_w^T M_1^T Q_1 M_1 X_w = 0 \\ W_2(x_w, y_w, z_w) = X_w^T M_2^T Q_2 M_2 X_w = 0 \end{cases} \tag{6}$$

The 3D conic curve S in space is the intersection curve of two cones W_1 and W_2.

2.3 Establishment of 3D Conic Curve

A 3D conic curve can be obtained by intersecting two cones. The intersection of two cones W_1 and W_2 of degree 2 is a planar curve lying on the plane p: $rx_1+ sy_1+ z_1+ t=0$, then there exist a scalar factor k and a polynomial f_c of degree1 such that,

$$W_1\text{-}kW_2=(rx_1+sy_1+z_1+t)\,f_c \tag{7}$$

According to this proposition, the polynomial W_1-kW_2 has a linear factor $rx_1+sy_1+z_1+t$. Now the problem of reconstructing the plane p becomes a multivariable polynomial factorization problem. In addition, k in the polynomial W_1-kW_2 is unknown. The representations of two cones W_1 and W_2 are

$$W_1=\begin{bmatrix}x\\y\\z\\1\end{bmatrix}^T\begin{bmatrix}a_{11}&a_{12}&a_{13}&0\\a_{12}&a_{22}&a_{23}&0\\a_{13}&a_{23}&a_{33}&0\\0&0&0&0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}=0\quad W_2=\begin{bmatrix}x\\y\\z\\1\end{bmatrix}^T\begin{bmatrix}b_{11}&b_{12}&b_{13}&b_{14}\\b_{12}&b_{22}&b_{23}&b_{24}\\b_{13}&b_{23}&b_{33}&b_{34}\\b_{14}&b_{24}&b_{34}&b_{44}\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}=0$$

The six equations provided by their resultant are

$$(a_{33}-kb_{33})r^2-2(a_{13}-kb_{13})r+a_{11}-kb_{11}=0 \tag{8}$$

$$(a_{33}-kb_{33})s^2-2(a_{23}-kb_{23})s+a_{22}-kb_{22}=0 \tag{9}$$

$$(a_{33}-kb_{33})t^2-(a_{33}-kb_{33}-2kb_{34})t-kb_{44}=0 \tag{10}$$

$$2(a_{33}-kb_{33})tr-2(a_{13}-kb_{13})t-(a_{33}-kb_{33}-2kb_{34})r-2kb_{14}=0 \tag{11}$$

$$(a_{33}-kb_{33})rs-2(a_{23}-kb_{23})r-(a_{13}-kb_{13})s+a_{12}-kb_{12}=0 \tag{12}$$

$$2(a_{33}-kb_{33})st-2(a_{23}-kb_{23})t-(a_{33}-kb_{33}-2kb_{34})s-2kb_{24}=0 \tag{13}$$

The variables r, s, and t are well separated in the first three equations. If k can be firstly solved for, then r, s, and t can be independently solved for by these one-variable quadratic equations. Since W_1-kW_2 is a quadratic polynomial and is reducible, it is well known that one of its invariants, D, defined by

$$D=|A-kB|=\left\|\begin{bmatrix}a_{11}&a_{12}&a_{13}\\a_{12}&a_{22}&a_{23}\\a_{13}&a_{23}&a_{33}\end{bmatrix}-k\begin{bmatrix}b_{11}&b_{12}&b_{13}\\b_{12}&b_{22}&b_{23}\\b_{13}&b_{23}&b_{33}\end{bmatrix}\right\|=0 \tag{14}$$

should be equal to zero. Therefore k is the eigenvalue of the matrix $B^{-1}A$.. After solving for k, we can solve for r, s, and t independently by equations (8) to (10). Since each equation has two solutions, we have eight solutions of (r, s, t) but some of them are not consistent with equations (11) to (13) and can be discarded. It is evident that the explicit parameters of the conic curve in space are completely defined by the cone W_1 and the plane p. After solving for p, we can obtain these parameters by simple algebraic operations.

Consequently, a 3D conic curve can be specified by the cone W_1 and the plane p, which can represent deformable human body occluding contours.

2.4 Establishment of Human Skeleton

The skeleton is a stick-model that represents the pose of the person in the image and makes it possible to segment the person into different parts. The motion of joints provides the key to motion estimation and recognition of the whole skeleton. The human skeleton system is treated as a series of jointed links(segments), which can be

modeled as a articulated body. For the specific problem of recovering the motion of a human figure, we describe the body as a stick model consisting of a set of fifteen joints (plus the head) connected by fourteen body segments [10], as shown in Figure 3(a). A closer look at the right arm is shown in Figure 3(b).

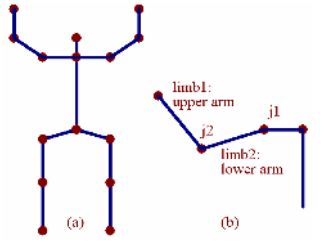


Fig. 3. a) The human skeleton b) A closer look at the right arm

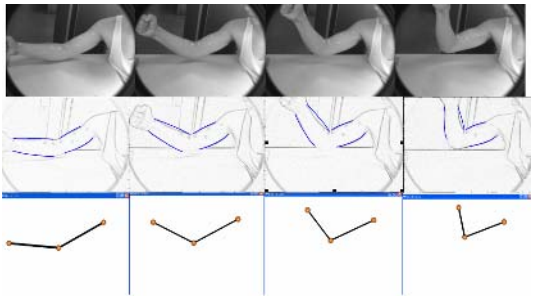


Fig. 4. The movement of human arm

3 Human Arm Parameters Determination

3.1 Deformation Parameters Determination of Human Body Occluding Contours

The image of a solid bounded by a smooth surface is itself bounded by an image curve, called the contour, silhouette or outline of this solid. This curve is the intersection of the retina with a viewing cone whose apex coincides with the pinhole and whose surface grazes the object along a second curve, called the occluding contour, or rim (see Fig.2).

We employ the gradient weighted least-squares estimation to solve the conic curve fitting[11].This method has the very attractive advantage that the eigenvector solution assures the global minimisation in case of convergence. We express deformation of human body occluding contours as the 3D conic curves of Section2.3. We fit one 2D conic curve to a set of points (x_i, y_i), lying on the contour of each limb in left or right images. A 2D conic curve can be expressed in the implicit form:

$$C(x,y) = (a_0 + a_1 x + a_2 y)(b_0 + b_1 x + b_2 y) - \rho(u_0 + u_1 x + u_2 y)^2 = 0 \tag{15}$$

Supposing $\{(x_i, y_i)\ , i \in 1, \cdots, n\}$ to be the contour points observed, the squared distance of a point (x_i, y_i) to the conic curve can be approximated by

$$dist\{(x_i, y_i), C(x,y)\}^2 \approx \frac{f^2(x_i, y_i)}{\|\nabla f(x_i, y_i)\|^2} \tag{16}$$

The gradient weighted least-squares criterion for fitting the function to the data set $\{(x_i, y_i)\}$ minimizes the mean square distance, denoted by the cost function

$$\Theta = \frac{1}{n}\sum_{i=1}^{n} dist\{(x_i, y_i), C(x,y)\}^2 = \sum_{i=1}^{n} w_i C^2(x_i, y_i) \tag{17}$$

where $w_i = 1/\|\nabla C(x_i, y_i)\|^2$ is the "gradient weight".

We minimize the function Θ to determine the best values of the deformation parameter ρ that describes the model's deformation. That is, we wish to determine the values of the deformation parameter that will make the conic curve pass through the given points data set with minimum error.

We use a stereoscopic vision system. There are two images of each stereo image pair, left and right images. Hence we can get four deformation parameters $\rho_{i\cdot left.upper}$, $\rho_{i\cdot left.lower}$ and $\rho_{i\cdot right.upper}$, $\rho_{i\cdot right.lower}$ of each limb at a time, that can represent the deformation of human arm properly.

3.2 Motion Parameters Estimation of Human Skeleton

We pick up the centers of the circles that pass through the vertices of the conic curve and are tangent to the boundary of each limb. We regard the points, lying on the sticks that linking these centers, as the skeleton points. It is well known that rigid motion can be expressed as a rotation R around the joint of the 3D coordinates followed by a translation T [12]. See Fig.3, the motion of the limb l_i is expressed as rotating around the joint j_i, then translating. j_i' is the joint after translation. R_i is the rotation matrix of the limb l_i in the 3D world coordinate system. $p_k \leftrightarrow p_k'$ are the correspondent 3D skeleton points on the limb l_i. Now, we can estimate the joints j_i, j_i' and the rotation matrix R_i over two image views by solving the following system of equations

$$p_k' = R_i(p_k - j_i) + j_i' \qquad j_i' = j_i + T_i \tag{18}$$

where $k = 1, \cdots, m$, m is the number of skeleton points on the limb l_i. The motion of the limb l_i can be estimated with only three skeleton points correspondences. We can solve the equation (18) with three skeleton points correspondences.

4 Experiment Results

The controlled scenes are acquired by using a stereoscopic vision system. The images used in the experiment are of an indoor scene containing a posing human arm(See Fig.4). The image sequence contains 10 stereo image pairs. However, only four pairs image frames from two cameras are used in this experiment since they show the deformation of body occluding contours typically. The human arm is nicely distinguishable against the simpler background using a combination of edge detection, greyscale, and intensity adjustment. The 2D coordinates of a set of points, lying on the contour of each limb and acquired from the image planes, are used as the input data for determining the deformation parameters of the limbs occluding contours. The gradient weighted least-squares estimation is employed to solve the conic curve fitting. the parameters of the plane, 3D conic curve lying on, are determinated by the method of Section 2.3.

The estimated results are listed in Table 1, Table 2 and Table 3 that describes the model's deformation and motion. The determined deformation parameters from left camera are given in Table 1.The determined plane parameters are given in Table 2. The estimated motion parameters are given in Table 3.

Table 1. Estimated deformation parameters

Deformation parameter	Frame1	Frame2	Frame3	Frame4
$\rho_{1.left.upper}$	16143	14389	846.6720	21.9888
$\rho_{1.left.lower}$	1.3200	1.2825	0.7877	-0.0616
$\rho_{2.left.upper}$	-8305.8	378.4237	11.9258	-0.4749
$\rho_{2.left.lower}$	109.2571	56.3748	2772.9	413.6866

$\rho_{1.left.upper}$ is the deformation parameter of the upper arm upper contour, $\rho_{1.left.lower}$ is the deformation parameter of the upper arm lower contour, $\rho_{2.left.upper}$ is the deformation parameter of the lower arm upper contour, $\rho_{2.left.lower}$ is the deformation parameter of the lower arm lower contour.

Table 2. Estimated parameters of plane p

	upper arm (r, s, t)	lower arm (r, s, t)
Frame1	(4.9723,8.5553,-412.8744)	(-1.5019,28.8184,166.2452)
Frame2	(4.2509,7.9248,-386.5332)	(0.8708,-3.8621,-1011.6)
Frame3	(3.6959,7.7719,-302.3469)	(0.2075,-1.7083,-936.8942)
Frame4	(2.6618,6.5545,-268.4768)	(-0.8521, -1.3332,-953.3770)

Table 3. Estimated motion parameters

(a) Upper arm (UA)

Time interval	Rotation axis $\vec{n}_1$			Rotation Angle θ_1	Translation $\overline{T}_1$		
	n_1	n_2	n_3		X	Y	Z
1—2	-0.6277	-0.7728	0.0938	-0.1852	7.5911	1.5811	-0.0837
2—3	-0.6462	-0.7576	0.0919	-0.3427	2.6562	-0.7987	2.5937
3—4	-0.9358	-0.3509	0.0353	-1.1869	-1.1896	-4.2581	9.1290

(b) Lower arm (LA)

Time interval	Rotation axis $\vec{n}_2$			Rotation Angle θ_2	Translation $\overline{T}_2$		
	n_1	n_2	n_3		X	Y	Z
1—2	-0.6419	0.4326	-.9947	0.1403	0.9377	13.5415	29.9642
2—3	0.8088	-0.5181	-.2782	-0.5021	-4.6424	18.0008	64.0877
3—4	0.6671	0.5096	-.5434	-0.7036	-7.8553	24.9998	236.7208

$\vec{n}_1, \theta_1$ and T_1 are the rotation axis, rotation angle and translation of the upper arm, $\vec{n}_2, \theta_2$ and T_2 are the rotation axis, rotation angle and translation of the lower arm.

Fig.4 depicts the movement of human arm. The first row are the original images of bare arm from left camera. The second row are the corresponding target contours and conic curve fitting results. The third row are the estimated 3D skeleton model.

5 Conclusion

We have presented a new method of elastic articulated objects (human bodies) modeling based on a new 3D conic curve. Our research focuses on two major points: 3D human body modeling and its parameters determination.Our experiments have demonstrated that our model can express the deformation of human body occluding contour properly. As future work we want to model other parts of human body, such as upper leg, lower leg and torso, from image sequence and determine their deformation parameters in 3D.

References

1. J. K. Aggarwal, Q. Cai and W. Liao.Articulated and elastic non-rigid motion: a review.Proc. IEEE Workshop On Motion of Non-rigid and Articulated Objects, 1994, pp. 2-14.
2. Monique Nahas,Herve Huitric,and Michel Saintourens.Animation of a B-Spline Figure. Visual Computer,1988.
3. Alex Petland and Bradley Horowitz.Recovery of Nonrigid motion and structure.IEEE Transaction on PAMI 1991,13(7):730~742.
4. Kyung-Ha Min, Seung-Min Baek, Gun A. Lee, Haeock Choi, and Chan-Mo Park. Anatomically-based modeling and animation of human upper limbs. in Proceedings of International Conference on Human Modeling and Animation, 2000.
5. Ralf Plankers and Pascal Fua. Articulated Soft Objects for Multiview Shape and Motion Capture. IEEE Transaction on PAMI, 2003, 25 (9): 1182 ~ 1187.
6. ESTIMATION ALGORITHMS FOR AMBIGUOUS VISUAL MODELS (C.Sminchisescu), Doctoral Thesis, INRIA, July 2002.
7. D'Apuzzo N, Plänkers R, Gruen A, Fua F and Thalmann D, Modelling Human Bodies from Video Sequences, Proc. Electronic Imaging1999, San Jose, California, January.
8. R.C.Jain, R.Kasturi and B.Schunk, Machine Vision, McGraw-Hill Inc, New York, 1995.
9. Songde MA.Conics-Based Stereo, Motion Estimation, and Pose Determination. Intern. J. Computer Vision, Vol. 10, No.1, 1993.
10. Xiaoyun Zhang, Yuncai Liu and TS Huang. Articulated Joint Estimation from Motion Using Two Monocular Images. Pattern Recognition Letters 25(10): 1097-1106,2004.
11. Z. Zhang. Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting. Image and Vision Computing Journal, 1996.
12. Xiaoyun Zhang, Yuncai Liu and TS Huang. Motion Estimation of Articulated Objects from Perspective Views. Lecture Notes in Computer Science, Vol. 2492, pp.165-176, 2002.

Discrete Conformal Shape Representation and Reconstruction of 3D Mesh Objects

Hongdong Li[1,2], Richard Hartley[1,2], and Hans Burkhardt[3]

[1] Research School of Information Sciences and Engineering,
The Australian National University
[2] ASSeT, Canberra Research Labs, National ICT Australia
[3] University of Freiburg, Computer Science Department, Freiburg, Germany

Abstract. This paper studies shape representation of general 3D objects. In particular, it proposes a conformal representation for genus-zero mesh objects, by using the discrete conformal mapping technique. It also proposes a new method to reconstruct the original shape from its conformal representation. In order to simplify and robustify the computation, we made several improvements to the above two procedures. The modifications include planar graph drawing initialization, Moebius factorization and spring-embedding-based reconstruction, etc. Though being mostly incremental, these modifications do provide significant improvements on previous methods. Possible applications include 3D geometry compression and object classification/recognition, etc.

1 Introduction

3D object representation and recognition is one of the central topics in computer vision and pattern recognition research. A good *shape representation* scheme is at the heart of any practical shape recognition systems. This paper aims at developing a new 3D shape representation method for general mesh objects.

We intend to derive *complete* representation. By *complete* we mean, such representation must fully encode all the necessary information of the original shape. As the result, it should be possible to recover the original shape from the representation(up to some approximation). In mathematical sense, this is equivalent to finding a mathematical representation of the original geometric entity (i.e., the shape).

For closed genus-zero mesh object, the shape representation problem effectively reduces to a *surface parameterization* problem. Discrete conformal mapping (DCM) is a newly developed surface parameterization technique in computer graphics and geometric processing areas[6][9][5]. Though the underlying mathematical principles of conformal mapping were well known over a century ago, how to apply them to modern digital mesh surfaces is still unclear to most practitioners. Conformal mapping has many nice properties that make it especially suited to the application of surface parametrization. The most notable one is that it preserves angles, and therefore preserves local geometry. In addition, it depends only on surface geometry (the Riemannian metric), and therefore is very robust to changes of data triangulations, resolutions (level-of-detail) and noise.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 535–542, 2005.

Our work in this paper basically follows [5] proposed by Gu and Yau et.al. It proposed a steepest-descend based iterative algorithm for global conformal parameterization of arbitrary genus objects, and presented many nice numerical results. It also showed the possibility for general 3D shape classification using conformal invariants[10].

However, when consider the application of 3D shape representation, their method have shortcomings. In particular, the mapping result may depend on human interactions, and the converge rate is rather slow. In order to better enjoy the nice properties of DCM while avoid most of the difficulties in its computation, we provide several necessary modifications in order to overcome most of these problems. Our new method is more efficient, and can automatically (no user interaction) produce complete shape representation. For demonstrating the completeness, we also proposed a shape reconstruction technique, which is used to reconstruct the original shape from its conformal representation. We have tested our method on a small set of mesh objects of different classes and complex geometries, and good results are obtained.

2 Discrete Conformal Mapping

Given two closed regular surfaces M_1 and M_2. According to the celebrated Riemann Mapping Theorem, for any two genus-zero surfaces there always exist conformal mappings between them. Therefore, a valid spherical parametrization of any genus-0 closed surfaces can always be found by such conformal mapping. However, the results are not unique (in fact, they are infinitely many). Nevertheless, all the feasible solutions actually form a low-dimensional subspace which is the Möbius group of 6-(real) parameters:

$$\mathbf{M}(z) = \frac{az+b}{cz+d}, ad - bc \neq 0, a, b, c, d \in \mathbf{C} \tag{1}$$

In practice, the conformal mapping is often approximated by a harmonic mapping, denoted by *f*. Namely, it must satisfy the following harmonic (Laplace) equation: $\triangle f = \mathbf{div}\ \mathbf{grad} f = 0$. For three-dimensional genus-0 surfaces, these two mappings are essentially the same. Therefore, the problem of finding a spherical conformal parametrization for genus-0 surface is reduced to a Laplace-on-Manifold problem, where the target manifold is the unit sphere S^2. Usually this is implemented by minimizing the following harmonic energy ([9][6]):

$$f = \arg\min_{f} E_H(f) = \frac{1}{2}\int_{M_2} \|\mathbf{grad} f\|^2 \tag{2}$$

3 Our New DCM Algorithm

Various methods for computing conformal mapping or discrete (triangulated) objects have been proposed. Our method basically follows paper[5]. To adapt this method for better fitting the 3D recognition purpose, we made several important improvements: (1) We introduce a new initialization method based on planar graph-drawing which effectively save many computations, and alleviate many fold-over problems. (2) We use the exponential map for solving Laplace-on-manifold diffusion problem, thus enlarges

the valid area of neighborhood and improves the convergence. (3) We introduce an affine stratification algorithm for the Möbius-normalization. This algorithm is simple, effective and much faster than other existing algorithms. In the following part we will briefly describe these modifications. (For more details see [8].)

3.1 Initialization from Planar Graph Drawing

We start from a triangulated closed mesh object. We assume it is topologically valid, namely, a closed manifold surface, no isolate element exists. There are several softwares publicly available for such topological check. We assume the mesh has spherical topology, which can be simply verified by Euler's formula,say, test whether $V - E + F = 2$. The minimization algorithm of the harmonic energy is iterative. It therefore requires a good initialization which serves as the starting point. This initialization should be a spherical homeomorphic approximation of the final mapping. Paper [5] provided an initialization method using Spherical Tutte Embedding , where the Tutte Embedding itself is started from a Gauss map. However, though theoretically it has good convergence property, we find that it often fails to converge correctly for complex meshes.

Based on the fact that the connectivity(adjacent) graph of any genus-0 object is always a *planar graph*, where by definition a *planar graph* is graph that can be drawn on a plane in such a way that there are no edge crossings, we propose a simple method for the spherical initialization. Since our diffusion algorithm has a relatively large neighborhood, it does not require an accurate initialization, so long as the homeomorphism is guaranteed.

There exist a number of constrictive algorithms that are able to actually draw a graph on a plane without edge crossing. Such is obvious a homeomorphism of the original mesh. In fact, every planar graph can be drawn such that each edge is straight, so-called *straight-line planar embedding*. Moreover, very efficient linear time algorithm for straight line embedding are also available now.

Our initialization procedure is: first arbitrarily select one surface triangle as the boundary triangle, then apply a straight-line planar graph drawing on the whole graph, and followed by an inverse stereographic mapping to get our spherical initialization result. Figure-1 illustrates an example of such planar embedding of a wolf mesh. Although the result depends on specific choice of boundary triangle, this speciality, however, will soon be relaxed by the subsequent diffusion process.

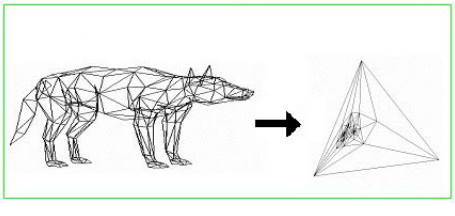


Fig. 1. A Planar graph drawing result of a wolf mesh

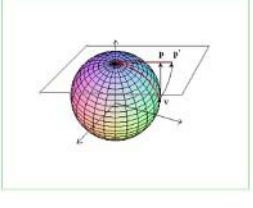


Fig. 2. Orthogonal map $\mathbf{p}$ and Exp map $\mathbf{p}'$ of the vertex $\mathbf{v}$

3.2 Harmonic Diffusion with the Exp-map

Having a homeomorphic spherical embedding as the initialization, the next step is to diffuse it to become a true conformal mapping. We accomplish this by solving a Laplace-diffusion equation on the unit sphere, namely, a Laplace-Beltrami Equation: $\triangle_{M_2} f = \mathbf{div}_{M_2} \mathbf{grad}_{M_2} f = 0$. Note that the Laplace operator has been adapted with respect to local geometry of the target manifold. Instead of directly solving this Laplace-Beltrami equation in Cartesian coordinates, which could be very involved, we adopt the use of tangent-plane-projection method. By this method the Laplace equation remains in its simple form, but acting on the tangent planes.

The purpose of tangent-plane-projection is to construct local coordinates systems on the manifold. Every tangent plane projection forms a local approximation, and can be regarded as a local representation at a local neighborhood (called a chart). Different mapping methods have different definitions of local neighborhood. Very often we would prefer a larger neighborhood definition, because by which we are able to use a less number of charts to approximate the whole manifold.

Orthogonal map is a simple method for tangent plane projection, and was adopted for DCM. However, acting on many manifolds it has a relatively smaller neighborhood compared with other methods. By orthogonal map, when two neighboring vertices are further apart than $\pi/4$ then they could not be included in a single chart.This will cause problem in computation.

We suggest the use of exponential map (exp-map, in short) to rectify this problem. The exp-map on manifold intuitively corresponds to expanding geodesic curve to tangent plane (See figure-2). It is easy to verify that for unit sphere the neighborhood of an exp-map is as large as π. In fact, this area can be further doubled if counting the direction of flow vectors. This means: by exp-map the valid neighborhood is as large as the whole sphere, which implies that all mesh vertices can find one-to-one maps on a single chart. This will ease the diffusion process, have better chance of convergence, and less depend on initial approximation. The computation of such exp-map on the unit sphere is also very simple thanks to the well-known Rodrigues formula.

3.3 Affine Factorization for Möbius Normalization

The solutions of conformal mapping from a surface to sphere are not unique. Simply applying another arbitrary automorphic conformal mapping to a solution will yield another valid solution. For the purpose of shape representation, we must ensure the uniqueness of the solution by using some *normalization* procedures.

Paper [5] suggested a nested-iteration algorithm for simultaneous diffusion and normalization. However, the required computations are extremely expensive especially for large scale meshes. Gotsman [9] use anchor point to normalize the solution, but the result depends on particular choice of the anchor point.

Though not unique, all the solutions actually have a relatively simple structure, say, all solutions follow a same Möbius transformation. They form a well-structured six dimensional *Möbius group*. Based on this important fact, we derive our new normalization algorithm, which significantly outperforms existing normalization algorithms.

Our method is based the concept *stratification*[2][3]. The main idea is to relax a fully Euclidean reconstruction to more general affine case, or projective case.

A Möbius Transformation has six degree-of-freedoms, which is a supergroup of the Euclidean group. We decide to decompose it into simple component transformations by analyzing its fixed points. It is easy to verify a Möbius group has at most two fixed points. If we keep the point at infinity invariant then we get an affine transformation, and the remaining one(i.e., the quotient) is proven to be a 3D rotation along the origin, which keeps the antipodal points remaining antipodal. This idea can be precisely clarified by the following operation.

Any Möbius Transformation can be represented by a 2x2 non-singular complex matrix $\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. Apply QR decomposition to this matrix, and after some algebras we get:

$$\mathbf{M} = \mathbf{Q} \cdot \mathbf{R} = \begin{bmatrix} q_1 & q_2 \\ -\overline{q}_2 & \overline{q}_1 \end{bmatrix} \cdot \begin{bmatrix} |r_1| & r_2 \\ 0 & |r_3| \end{bmatrix} \tag{3}$$

It is easy to verify that the orthonormal factor $\mathbf{Q}$ is a *quaternion* that precisely describes the 3D rotation along the origin, and the upper triangular factor $\mathbf{R}$ is indeed an affine transformation. Both factors has three degree-of-freedoms(not counting an arbitrary real scale factor), so each of them is a three-parameter subgroup.

Our strategy is to find a special affine factor $\mathbf{R}$ such that the following equation(which is obvious an invariant wrt. the 3D rotation) is satisfied:

$$\int_{S^2} \left[\phi^{-1} \circ \mathbf{R} \circ \phi\right] \circ f \, \mathbf{d}\sigma_{M_1} = 0 \tag{4}$$

where φ is the stereographic mapping $\varphi(x, y, z) = \left(\frac{x}{1-z}, \frac{y}{1-z}\right), x, y, z \in S^2$, $\mathbf{d}\sigma_{M_1}$ is the area-element on the original shape. Since the *f* represents coordinates of vertices on target manifold, there are actually three equations. Now, our Möbius normalization procedure is reduced to very small-scale equations with the three *real* unknowns of the factor $\mathbf{R}$. Once solved, apply the corresponding spherical affine transformation $\mathbf{R}$ will give us a unique solution up to rotation.

4 Reconstruction from Spring Embedding

Now we obtain the spherical conformal parameterization of the input mesh (,for example, see figure-2). The next step will be to represent the original shape on this sphere parameterization. First we need to specify functions on the sphere, these functions themselves should be complete in the sense they can faithfully represent the original shape without information loss.

Theorem: *A closed surface $r(u, v)$ in $\mathbf{R}^3$ with parameter (u, v) is determined by its first fundamental forms and its mean curvature $H(u, v)$ uniquely up to rigid motions.*

From the above theorem(the proof can be found elsewhere in a differential geometry text book), it is clear that the surface can be reconstructed from the first fundamental

form and the mean curvature. Following [11], we also use the edge length and dihedral angle as the shape functions, because the first fundamental form is represented as the length of edges, the mean curvature is represented as dihedral angles of edges. Since these shape functions are complete, in turn it should be possible to uniquely (up to rigid motions) reconstruct the original surface from the two set of data.

Paper [11] suggests a reconstruction method, which is based on the solving of a set of simultaneous equations of local distances. In noise-free case this method works fairly good. However, when there is even small noise (for example, due to numerical precision), this algorithms may give rise to a very distorted shape. We here propose

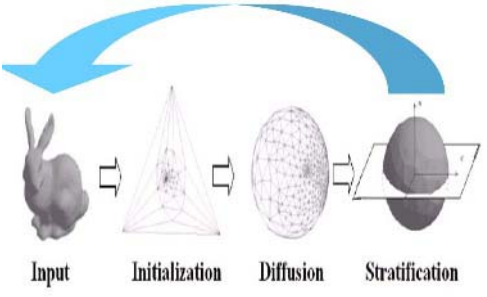


Fig. 3. Conformal shape representation and reconstruction. (The arrow above shows the direction of the reconstruction)

a global shape reconstruction method. This method is based on 3D graph drawing. In particular we apply a *Spring Embedding* algorithm for drawing a graph in 3D.

The spring embedding is a conceptually simple yet powerful technique for drawing 3D graph. For our application, the graph to be drawn is actually a planar graph (section 3-1). What we have now are both of its local distances (i.e., the edge lengthes) and second order distances. (The latter can be computed easily from the dihedral angles, see [4]). The total Springs Energy is thus written as:

$$W = \sum_{i=1}^{n-1} \sum_{j \in N_i} \frac{1}{2} k_{ij} \left(|v_i - v_j| - l_{ij} \right)^2 \tag{5}$$

where the v_i, v_j are vertex coordinate vecteros, l_{ij} represent both the edge lengthes and second order distances, n is the total number of vertices, N_i represents the up to second order nationhood. k_{ij} is the spring constant, here we set it to the inverse of the corresponding distance.

We use a Gauss-Newton method to solve this minimization problem of eq 5. Experiments show that the convergence is very fast, the iteration finishes within 10 steps for a mesh object of about 2,000 edges. The recovered shapes are almost identical to the original ones.

5 Experiments and Results

We tested our algorithm on a set of mesh objects of difference classes. We first performed topological validation on them. For those that do not have valid spherical topologies we manually modified them. For example, some holes on the bottom of the Stanford bunny model had been filled.

Figure-4 shows examples of the DCM mapping results of different objects by our algorithm. The spheres showed in the right column are the result after Möbius normalization (affine stratification), so they are unit up to rotation. We have positively verified this by applying the algorithm again on a randomly rotated object.

We also tested the 3D reconstruction method based on spring embedding. Figure-5 show some results, the value of W gives the spring energy after converge. Figure-6 gives an example convergence curve for Bunny mesh.

We demonstrated the robustness of our methods with respect to different triangulations, resolution and different noise. We performed both subdivision-based refinement and edge-collapse-based simplification operations to the original meshes, and obtained same object with different triangulations and resolution. We further introduced isotropic Gaussian noise to the vertices coordinates (in the DCM stage) and to the edge distances(in the reconstruction stage), then apply our algorithms again on these distorted meshes. The new results are still very stable (for space limit, we have not present the results here), which indicates the methods are robust.

Fig. 4. DCM mapping results by our method

6 Conclusions

We have proposed a conformal method for representing arbitrarily-complex genus-zero 3D mesh objects. We have also demonstrated the possibility of reconstructing the shape from its representation. This is only the first step toward describing more complex and more general 3D objects (for example, objects with higher-genus). It is expected that the

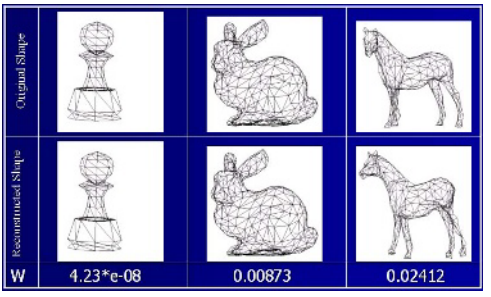


Fig. 5. Shape reconstruction from spring-embedding. (W is the final springs energy.)

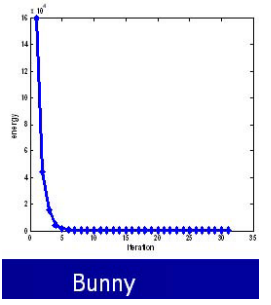


Fig. 6. A convergence curve. (energy .vs. iterations)

proposed method can find many practical applications, such as 3D geometry compression [1]and shape recognition[7][10]. For this purposes, more and much harder work still need to be done.

Acknowledgments. National ICT Australia is funded through the Australian Government's Backing Australia's Ability Initiative, in part through the Australian Research Council. Thanks given to Fredrik Kahl in many insightful discussions. T.B.Chau has helped with the programming of spring embedding. The mesh objects used in experiments were downloaded from C. Grimm's web site (http://www.cs.wustl.edu/ cmg/).

References

1. G.Taubin, J. Rossignac, Geometric Compression Through Topological Surgery, ACM Transactions on Graphics,Vol.17, no.2, pp84-115,1998.
2. O. Faugeras, Stratification of 3-D vision: projective, affine, and metric representations, JOSA-A,1995.
3. R.Hartley, and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd-edition, Cambridge University Press,2004.
4. H.Burkhardt,M.Reisert,H.Li, Invariant for Discrete Structures-An Extension of Haar invariant over Dirac Delta Functions, Pattern Recognition Symposium, DAGM-2004, Germany, 2004.
5. X.Gu, Y.Wang, Tony F.Chan, Paul M.Thompson, S.Yau. Genus Zero Surface Conformal Mapping and Its Application to Brain Surface Mapping. IEEE T-Medical Imaging, VOL.23, NO.8, AUGUST 2004.
6. M. S. Floater and K. Hormann, Surface Parameterization: a Tutorial and Survey, *Advances in Multiresolution for Geometric Modelling*, pp157-186, Springer,2005.
7. H.Shum, M.Hebert, K.Ikeuchi, On 3D Shape Similarity,IEEE-CVPR-1996, 1996.
8. Hongdong Li, R.I. Hartley, A new 3D Fourier descriptor for describing arbitrary genus-0 mesh objects, RSISE/NICTA Technique Report, 2005.
9. C. Gotsman, X. Gu, A. Sheffer, Fundamentals of Spherical Parameterization for 3D Meshes, ACM Transactions on Graphics (TOG),Vol-22,issue-3,pp358-363, July 2003.
10. X. Gu, S. Yau. Surface Classification Using Conformal Structures. in Proc ICCV-2003, 2003.
11. X. Gu, Y.Wang, S.Yau, Geometry compression using Riemannian structure, Communication in information systems,Vol.3, No.3, pp171-182,2004.

Security Enhancement of Visual Hashes Through Key Dependent Wavelet Transformations*

Albert Meixner[1] and Andreas Uhl[2]

[1] Department of Computer Science,Duke University, USA
[2] Department of Scientific Computing,Salzburg University, Austria

Abstract. Parameterized wavelet filters and wavelet packet subband structures are discussed to be used as key dependent wavelet transforms in order to enhance the security of wavelet based hashing schemes. Experiments show that key dependency and keyspace of the hashing scheme considered have been significantly improved. The attack resistance could only be slightly enhanced by using parametrized wavelet filters.

1 Introduction

The use of robust hash functions for image authentication has become a popular field of research. A key problem in the construction of secure hash values is the selection of image features that are resistant to common transformations. To ensure the algorithm's security [7], these features are required to be key dependent and not computable without knowledge of the key used for hash construction. For example, the Visual Hash Function (VHF) [3] projects image blocks onto key dependent patterns to achieve this goal.

In recent work [5] we have shown a simple attack against a wavelet-based robust hashing scheme introduced by Venkatesan et al. [9]. In this context we have pointed out that a key-dependent parameterized wavelet transform could serve as a generic way to improve the security of such algorithms. A similar approach has been proven to enhance the security of various watermarking schemes, using both key-dependent filter parameterization [2] and key-dependent wavelet packet subband structures [1].

In this paper two different methods of adding key dependency to the wavelet transformation are proposed. In the experiments, these wavelet transformations are evaluated with respect to their sensibility to changes in the key material and the available keyspace when used in the context of the hahsing scheme of Venkatesan et al. [9]. Finally we test the usefulness of those schemes to counter the attack [5] against this algorithm.

2 Key-Dependency Schemes

Pseudo Random Partitioning. A common approach to generate secret image features is to first create a pseudo-random partitioning of the image and compute

* This work has been partially supported by the Austrian Science Fund FWF, project no. P15170.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 543–550, 2005.

features independently for every partition. The exact values of the features can not be computed without knowledge of the key used to seed the PRNG, because the regions on which the features are computed are not known.

Random paritioning is used as original key-dependency scheme in the hash algorithm of Venkatesan et al. [9]. Its use is orthogonal to the following two schemes and can be easily combined with either of them to further increase security (which will be done in our experiments).

Random Wavelet Packet Decomposition. In the classical wavelet transformation only the low-low-sub-band can be further decomposed, resulting in the typical pyramidal structure. Wavelet packet decomposition [1] removes this constraint and allows to further decompose any sub-band. The decision which sub-bands are decomposed is either determined by a given structure or based on some measure of optimality.

By using a pseudo random number generator to decide, if a sub-band should be further decomposed, we can make the decomposition structure key dependent. This approach has been shown to be effective in selective image encryption [6] and in securing watermarking schemes [1].

Parameterized Filters. Wavelet decomposition typically uses fixed, well known filters, such as the Daubechies filters. There are also methods to generate families of wavelet filters from a number of parameters, that can be freely chosen (we employ a familiy of parameterized orthogonal Daubechies wavelet filters [8]). If these parameters are kept secret, they can be used as a key for the decomposition. Similar to the wavelet packet case, this type of key-dependency has been used before in selective image encryption [4] and watermarking [2].

3 Experiments and Results

We have tested both proposed schemes by including them into a authentication hash algorithm introduced by Venkatesan et al. [9]. The original algorithm achieves key dependency through random partitioning. We use this algorithm as a base case:

- The image is transformed, using a 3-level pyramidal wavelet transformation
- For each of the resulting subbands a feature vector F_i is calculated. This is done by randomly partitioning the subband and calculating a statistical measure for each region.
  For the approximation the statistical measure used is the arithmetic average, for all other subbands the variance is computed.
- The real number elements of each F_i are projected to $\{0 \dots 7\}$ using randomized rounding. The resulting values are concatenated to form a preliminary hash string H_p.
- The hash string H_p is shortened by feeding it into the decode stage of a Reed-Muller error correcting code. This does not only shorten the hash string, but also improves robustness.

Table 1. Hamming distances among a set of images

	baboon	barb	boat	jet	lena	peppers	truck	zelda
baboon	0.00	0.43	0.46	0.32	0.35	0.32	0.37	0.38
barb	0.43	0.00	0.35	0.39	0.37	0.44	0.47	0.40
jet	0.32	0.39	0.39	0.00	0.31	0.40	0.48	0.34
lena	0.35	0.37	0.40	0.31	0.00	0.36	0.45	0.30

– In the final step a Linear Code algorithm is applied to the hash, again both shortening it and increasing robustness.

To obtain an initial estimate and upper bound of the Hamming distance threshold for considering an image untampered, a set of different images is compared.

The Hamming distance between two independent images is consistently below the optimal distance of $\frac{1}{2}$. This is mainly a result of the fixed values used in the randomized rounding procedure, which favor the lower and upper bounds, and a non uniform distribution of features values. For more detailed results and some improvements of the alorithm see [5].

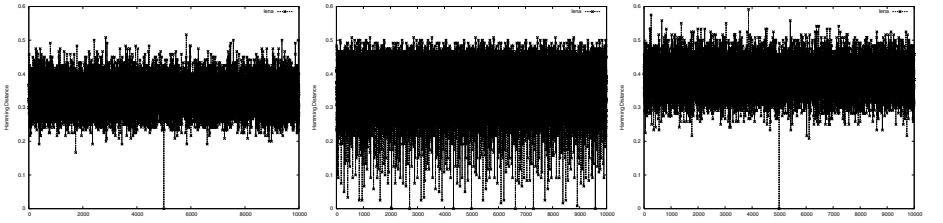
3.1 Key Dependency

A key dependency scheme can only improve security if the choice of the key has a significant impact on the resulting hash value. All following figures show the normalized Hamming distance of a hash created with some fixed key value to other hashes, produced with varying other key values. Key values are displayed along the ordinate, resulting Hamming distances along the abscissa.

The random partitioning approach, though vulnerable by a simple attack (see [5] and subsection 3.3), is very effective in adding key dependency, with average Hamming distance 0.336 and very few keys reaching values below 0.2 (see Fig. 1(a)). The figure shows the results 10000 different partitions, compared to a fixed key at position 5000. A similar phenonemon (i.e. security weaknesses in spite of a key-dependent hash) was pointed out by Radhakrishnan et al. [7] for the block-based VHF. This contradictory behaviour was improved by adding block inter-dependencies to VHF.

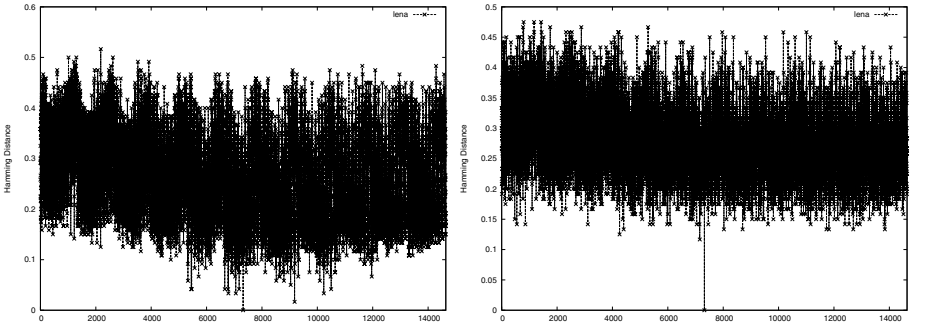
Random wavelet packet decompositions with a constant decomposition probability for all subbands makes shallow trees far more likely than deep trees. This increases the chance of collisions, especially for shallow trees. Following a previous suggestion [6], we use a higher initial decomposition probability for the first decomposition level and decrease it subsequently for every subsequent decomposition recursion (we use a base value of 0.9 ($p = 0.55$) and a change factor of -0.1 [6]). The obtained average Hamming distance (Fig. 1(b)) is 0.3570 and about 0.73% of all distances are below 0.1. However, we result in 20 "almost" correct keys (distance < 0.05) which makes the approach less reliable.

Even with random decomposition in place, the key of the standard algorithm required to create partitions for extracting localized feature vectors may



(a) Random partitioning (b) Random decomposition (c) Random decomposition & partitioning

Fig. 1. Key dependency test: Hamming distances between hashes generated with different keys



(a) Random filters (b) Random filters & partitioning

Fig. 2. Key dependency test: Hamming distances between hashes generated with different keys

be varied as well, thus increasing the key space and possibly overall security. Fig. 1(c) shows key dependency results for varying both keys. The average distance for this setup increases to 0.3884 with no incorrect keys reaching distances below 0.1. Combining both strategies obviously significantly increases the keyspace while maintaining the high sensitivity to key variations of the original standalone random partitioning scheme.

Experiments concerning filter parametrization are based on a parameterized filter with 4 parameters $(1.0, 1.5, -2.0, -1.0)$, all parameters were modified in a range of ± 1.0 in steps of 0.2, resulting in $11^4 = 14641$ combinations. The correct key for this test is 7320. The results for parameterized filters are almost as good as the random partition scheme, with an average of 0.265 and only 0.53% of the keys below 0.1 (see Fig. 2(a)).

Similar to the random decomposition, using parameterized filters adds key dependency to the decomposition stage. Thus, the parameterization key can also be combined with the standard partitioning key used during a later stage of the scheme. When both keys are used, the average hamming distance in-

creases slightly to 0.2795, additionally there are no more incorrect keys reaching values below 0.1 (see Fig. 2(b)). Again, combining the two schemes maintaines sensitivity towards key alterations while increasing the keyspace.

3.2 Key Space

A major concern of any key dependent algorithm is the number of distinct keys that can be used. If the number of keys is too small, the scheme is vulnerable to brute force attacks. The discrete key space of both random partitioning and random decomposition grows exponentially with a free algorithm parameter (e.g., following the formula given in [6], a decomposition depth of 5 leads to $\approx 2^{1043}$ different keys in random decomposition). Thus the size of the key space can be easily adjusted and it seems that a suitable number of keys is available for any level of security desired. However, a bigger number of keys may have some undesired side effect on the overall algorithm.

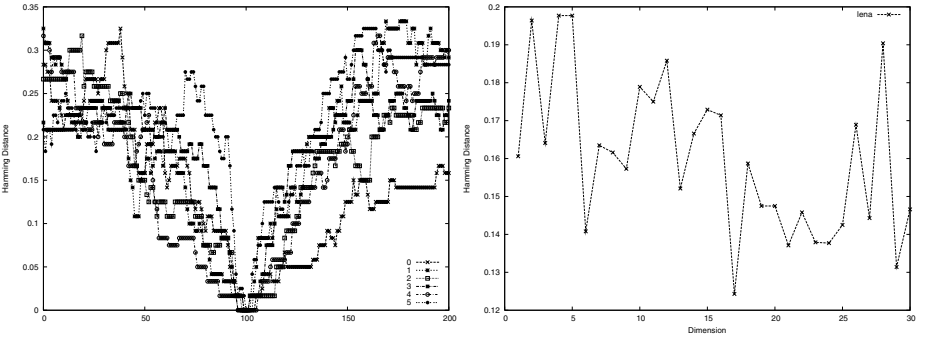
In random partitioning, the areas get smaller with an increasing number of keys. This makes the hash more sensitive to minor image modifications and many keys will produce fairly similar results. Random decompositions suffers from the fact, that high decomposition depth leads to a big number of very similar tree structures, which lead to identical hash values. Therefore, the keyspace needs to be set to some sensible compromise in this two cases (e.g. decomposition depth 5 is a good choice for random decomposition).

Contrasting to the previous cases, the key values are continuous rather than discrete for filter parametrization. Therefore, a quantization must be defined to determine the number of possible keys. This can be done by defining a range of valid parameters $(d_{min} \dots d_{max})$ and quantization function $Q(d) = \left\lfloor \frac{d}{q} \right\rfloor$. Now the the number of keys $f(n)$ for a filter with n parameters can be calculated: $f(n) = \left\lfloor \frac{d_{max} - d_{min}}{q} \right\rfloor^n$. The filter parametrization used is based on trigonometric functions (sin, cos). Thus, the parameters have a range of $(-\pi \dots \pi)$.

n	Keys	
1	125	$\approx 2^7$
2	15625	$\approx 2^{14}$
3	1953125	$\approx 2^{21}$
4		$\approx 2^{28}$
5		$\approx 2^{35}$
6		$\approx 2^{42}$
7		$\approx 2^{49}$
8		$\approx 2^{56}$
9		$\approx 2^{63}$

Table 2. Parameterized Filters Key Space

In the following, we determine the quantization function by a simple experiment. Fig. 3(a) shows the results, if only one parameter of a 6 dimensional parameterization is modified in the range of ± 1.0 with a step size of 0.01. There is a curve for every one of the six single parameters. The graph's values change in multiple steps, suggesting that key values within about 0.05 produce the same hash. Thus, when generating parameters from the key the granularity should be $0.05 - 0.10$ (the parameters used to create the graph were $(1.0, 1.5, -2.0, -1.0, 0.0, 0.5)$). To be on the safe side, we limit the the distance in a single parameter between two keys to be no smaller than 0.1. Using these values, the number of available keys can be calculated as: $f(n) = \left\lfloor \frac{\pi - (-\pi)}{0.1} \right\rfloor^n = \lfloor 20.0 \cdot \pi \rfloor^n \approx 62.8^n$. The number of resulting keys dependent on n is shown in Table 2.



(a) Varying one out of six parameters

(b) Average distances for varying the first parameter

Fig. 3. Hamming distances

The granularity q is very important for the security of the scheme and might be dependent also on the number of parameters n. It seems intuitive, that the influence of a single parameter on the overall result will decrease for a higher numer of parameters. This, however, is not the case as shown in Fig. 3(b). For every filter dimension shown on the x-axis, the average Hamming distance between the hash for a fixed parameter vector and all hashes resulting from the *first* parameter of this vector being changed in the range of ± 1.0 is shown on the y-axis. This average distance indicates how much influence a single parameter has on the resulting hash value – it varies significantly from 0.12 to almost 0.2 without any clear trend up or downwards for an increasing number of dimensions. Thus, d does not have to be selected dependent on n.

3.3 Attack Resistance

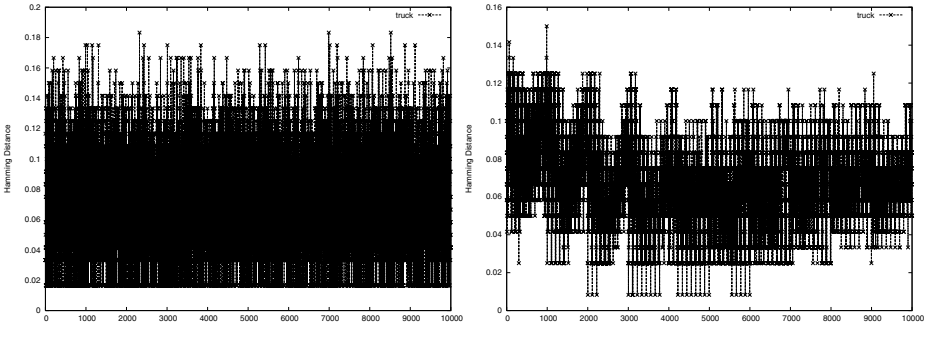
Fig. 4. Forged & attacked Lena image

The reason for the idea of enhancing the original partitioning scheme with a key dependent wavelet transformation is its vulnerability to the simple attack shown in [5]. The major problem of the use of variance and average as basis of the hash value is that both are publicly available and very easy to modify [3]. Both average and variance mostly change gradually within an image, so that if the measures of two images match within a certain partition, they will at least be similar within any other partition covering approximately the same area as well. This is exploited by the referenced attack.

Fig. 4 shows a forged and attacked version of the Lena image with a Hamming distance of 0.01 to the original. The image modification without attack mounted

exhibits Hamming distance 0.12 to the original which would have been detected as forgery of course. Since this value is significantly below the Hamming distance of that between the original and a JPEG compressed version with high quality, the picture would be rated identical to the original by the hashing algorithm. This example shows the severness of this attack drastically. See [5] for more details on the attack and corresponding results.

The goal of the proposed new schemes is to eliminate feature correlations between transformations computed with different key values. Though some parameters apparently result in the exact same hash value, overall hash values strongly depend on the selected parameters as we have seen in the previous subsections. Attempting an attack gets very hard without knowledge of the transform domain used for creating the original hash. The underlying assumption of the attack is, that it is operating on a transformed image identical to the one used to calculate the hash value. Only if this is the case, adjusting the forgery's features to match those of the original has the desired effect on the hash value. By using a tranform domain with an incorrect set of parameters, this assumption is weakened. The adjusted forgery's features will only match those of the original for the filter chosen for the attack. This does not necessarily make them more alike for any other filter. Fig. 5 shows the results of the attack using both techniques and various decomposition keys.



(a) Wavelet Packets (b) Parameterized Filters

Fig. 5. Attack resistance of the key dependency schemes

The Hamming distance for the correct key in the random decomposition case after the attack has been mounted is 0.0166. The average distance after the attack for all random decompositions considered is inceased to 0.0728, however, the large number of "correct" keys (i.e. leading to the same result as the key used to compute the original hash) makes the scheme unreliable (Fig. 5(a)). This corresponds well to the results with respect to key dependency displayed in Fig. 1(b).

Given the key dependency tests (Fig. 2(a)), filter parameterization seemes more promising than random decomposition. Though only a small number of

filters renders the attack completely useless, its effects are attenuated considerably, thus improving the scheme's overall security. The average distance of 0.0666 after the attack, compared to 0.0083 for the correct key, is a definite improvement (see Fig. 5(b)). The number of successful attacks (i.e. equally successful as without filter parametrization) is negligible. However, considering the high number of key values with still rather low Hamming distances, the effects of the attack can only said to be weakened to some extent.

4 Conclusion

We have discussed the use of key dependent wavelet transforms as a means to enhance the security of wavelet based hashing schemes. Whereas key dependency and keyspace of the hashing scheme considered in experiments have been significantly improved, the attack resistance has been improved by using parametrized wavelet filters to a small extent only.

References

[1] W. M. Dietl and A. Uhl. Robustness against unauthorized watermark removal attacks via key-dependent wavelet packet subband structures. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '04*, Taipei, Taiwan, June 2004.

[2] Werner Dietl, Peter Meerwald, and Andreas Uhl. Protection of wavelet-based watermarking systems using filter parametrization. *Signal Processing (Special Issue on Security of Data Hiding Technologies)*, 83:2095–2116, 2003.

[3] Jiri Fridrich and Miroslav Goljan. Robust hash functions for digital watermarking. In *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, USA, March 2000.

[4] T. Köckerbauer, M. Kumar, and A. Uhl. Lightweight JPEG 2000 confidentiality for mobile environments. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '04*, Taipei, Taiwan, June 2004.

[5] Albert Meixner and Andreas Uhl. Analysis of a wavelet-based robust hash algorithm. In Edward J. Delp and Ping W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306 of *Proceedings of SPIE*, pages 772–783, San Jose, CA, USA, January 2004. SPIE.

[6] A. Pommer and A. Uhl. Selective encryption of wavelet-packet encoded image data — efficiency and security. *ACM Multimedia Systems (Special issue on Multimedia Security)*, 9(3):279–287, 2003.

[7] R. Radhakrishnan, Z. Xiong, and N. D. Memom. Security of visual hash functions. In Ping Wah Wong and Edward J. Delp, editors, *Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents V*, volume 5020, Santa Clara, CA, USA, January 2003. SPIE.

[8] J. Schneid and S. Pittner. On the parametrization of the coefficients of dilation equations for compactly supported wavelets. *Computing*, 51:165–173, May 1993.

[9] Ramarathnam Venkatesan, S.-M. Koon, Mariusz H. Jakubowski, and Pierre Moulin. Robust image hashing. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'00)*, Vancouver, Canada, September 2000.

Conversion Scheme for DCT-Domain Transcoding of MPEG-2 to H.264/AVC

Joo-kyong Lee and Ki-dong Chung

Dept. of Computer Engneering, Pusan National Univ., Jangjeon-dong, Geumjeong-gu, Busan, 609-735, Korea
jklee@melon.cs.pusan.ac.kr, kdchung@pusan.ac.kr

Abstract. The 4×4 approximate discrete cosine transform (DCT) of H.264/AVC [1] makes it difficult to transcode the pre-coded video contents with the previous video coding standards to H.264/AVC in DCT domain. This is due to the difference between 8×8 DCT used previous standards and 4×4 DCT in H.264/AVC. In this paper, we propose an efficient algorithm that converts the quantized 8×8 DCT block of MPEG-2 into newly quantized four 4×4 DCT blocks of H.264/AVC to support DCT-domain transcoding. Experimental results show that the proposed scheme improves computational complexity by 5~11% and video quality by 0.1 ~ 0.5 dB compared with cascaded pixel-domain transcoding that exploits inverse quantization (IQ), inverse DCT (IDCT), DCT, and re-quantization (re-Q).

1 Introduction

As the number of networks, types of devices, and video representation formats increase, interoperability between different systems and different networks is becoming more and more important [2]. To provide a seamless interaction between producers and consumers, diverse research on video transcoding such as bit-rate reduction [3], spatial resolution reduction [4], frame skipping [5][6], and simple video format conversion [7], has been conducted. Recently, DCT-domain transcoding approaches have been studied to improve computational complexity and to avoid DCT and IDCT mismatch problem [8][9]. These approaches are based on 8×8 DCT used in most video coding standards such as MPEG-2, MPEG-4, and H.263/AVC. However, it is impossible to directly apply them to H.264/AVC, because it uses 4×4 transforms.

In this paper, for non-intra-coded blocks, we propose an efficient algorithm to convert an 8×8 DCT block of MPEG-2 to four 4×4 DCT blocks of H.264/AVC. Additionally, we propose a quantization conversion algorithm that changes quantization step size between the two standards. This algorithm improves video quality by reducing the quantization error caused by cascaded IQ/re-Q. However, for intra-coded blocks, we follow the cascaded re-encoding method in pixel domain instead of DCT conversion to avoid heavy computational complexity in prediction mode selection. Our scheme can be easily applied to other standards by slightly modifying the quantization conversion algorithm. This paper is organized as follows. In Section 2 we de-

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 551 – 558, 2005.

scribe 4×4 transformation for H.264/AVC, respectively. In Section 3 we present the Qstep conversion scheme and DCT conversion scheme. Experimental results will be presented in Section 4, and the conclusion is shown in Section 5.

2 Transformation for H.264/AVC [10] [11]

The 4×4 transformation of H.264/AVC approximates the 'true' 4×4 DCT. The 4×4 transform matrix, H of the 'true' DCT can be illustrated by Eq. 1, where a, b, c represent the matrix elements. H.264/AVC, On the other hand, uses the modified transform matrix H', where b and b/c are changed to $\sqrt{2/5}$ and $\frac{1}{2}$, respectively. In actual H.264/AVC implementations, the integer transform matrix is used for transformation, and the residual scaling factors are absorbed into quantization process to avoid multiplication operations, and maintain DCT/IDCT accuracy. See [10] as well as [11] for more detail. In this paper we use H' to convert the 8×8 DCT block into four 4×4 DCT blocks of H.264/AVC, and we cover the residual scaling factors using quantization step size (Qstep) conversion tool. Throughout this paper, we represent the result of the modified 4×4 DCT using H' as '4×4 approximate DCT'. Since there is a difference between 'true' DCT and 'approximate' DCT, we should compensate it when we convert the 8×8 DCT block into four 4×4 approximate DCT blocks.

$$Y = HXH^T, H = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}, where \;\; a = \frac{1}{2}, \;\; b = \sqrt{\frac{1}{2}}\cos(\frac{\pi}{8}), \;\; c = \sqrt{\frac{1}{2}}\cos(\frac{3\pi}{8}) \tag{1}$$

3 Proposed Conversion Schemes

In this section we present two conversion schemes, Qstep conversion scheme and the DCT conversion scheme. The Qstep conversion scheme changes the quantized DCT coefficients of MPEG-2 to newly quantized coefficients of H.264/AVC. This scheme covers the H.264/AVC quantizer that is associated with transformation by incorporating the residual scaling factors into it. DCT conversion scheme converts the 'true' 8×8 DCT into the 4×4 approximate DCT.

3.1 Qstep Conversion

In MPEG-2 encoder, intra coded macroblock (MB) and non-intra coded MB are quantized by different formula and quantizer matrices. Intra coded MB coefficients are quantized as shown in Eq. 2, where $\hat{B}_{(i,j)}$ refers to the $(i,j)^{th}$ coefficient of the DCT block $\hat{B}$, $Q(\hat{B}_{(i,j)})$ refers to the quantized value of $\hat{B}_{(i,j)}$, and q refers to Qstep value. The DC coefficients of the luminance and chrominance components of the intra coded MB are divided by one of the Qstep values 8, 4, 2 and 1. For AC coefficients, first, the coefficients are scaled by 32 and corresponding elements of the intra quantizer matrix $W_{(i,j)}^{intra}$, and secondly, they are quantized by Qstep value q ranging from 1 to 112.

$$Q(\hat{B}_{(i,j)}) = \begin{cases} \hat{B}_{(i,j)} / q \ \text{ for } \ i = j = 0 \\ \{(32 \cdot \hat{B}_{(i,j)}) + \text{sign}(\hat{B}_{(i,j)}) \cdot q \cdot W_{(i,j)}^{intra}\} / (2q \cdot W_{(i,j)}^{intra}) \ \text{ for all } \ i, j \text{ except } i = j = 0 \\ \text{, where} \quad \text{sign}(\hat{B}_{(i,j)}) = \begin{cases} 1, & \hat{B}_{(i,j)} > 0 \\ 0, & \hat{B}_{(i,j)} == 0 \\ -1, & \hat{B}_{(i,j)} < 0 \end{cases} \end{cases} \tag{2}$$

$$\hat{B}_{(i,j)} = \frac{q \cdot W_{(i,j)}^{intra}}{16} \{Q(\hat{B}_{(i,j)}) - \frac{\text{sign}(\hat{B}_{(i,j)})}{2}\} \tag{3}$$

$$\frac{\hat{B}_{(i,j)}}{q'} = \frac{q}{q'} \cdot \frac{W_{(i,j)}^{intra}}{16} \cdot \{Q(\hat{B}_{(i,j)}) - \frac{\text{sign}(\hat{B}_{(i,j)})}{2}\} \tag{4}$$

$$Q(\hat{B}_{(i,j)}) = (32 \cdot \hat{B}_{(i,j)}) / (2q \cdot W_{(i,j)}^{non\text{-}intra}), \ 0 \le i, j \le 7 \tag{5}$$

$$\hat{B}_{(i,j)} = \frac{q}{16} W_{(i,j)}^{non\text{-}intra} \cdot Q(\hat{B}_{(i,j)}) \tag{6}$$

$$\frac{\hat{B}_{(i,j)}}{q'} = \frac{q}{q'} \cdot \frac{W_{(i,j)}^{non\text{-}intra}}{16} \cdot Q(\hat{B}_{(i,j)}) \tag{7}$$

Now let us change Eq. 2 to transcode the quantized coefficients, $Q(\hat{B}_{(i,j)})$, to newly quantized values of H.264/AVC. First, the DC coefficients are consecutively dequantized by q and then re-quantized by new Qstep q'. Secondly, the unquantized AC coefficients, $\hat{B}_{(i,j)}$, of MPEG-2 as described in Eq. 3, derived from Eq. 2, can be calculated the newly quantized coefficients using q' as described in Eq. 4. Briefly, using Eq. 4, we convert quantized AC coefficients of MPEG-2 into newly quantized coefficients of H.264/AVC. This method has an effect of skipping MPEG-2 quantization. That is, we can calculate the newly quantized coefficients of H.264/AVC as if we performed quantization only once with the unquantized coefficients, $\hat{B}_{(i,j)}$ without any quantization for MPEG-2. As a result, the proposed Qstep conversion scheme can reduce the dequantization errors of MPEG-2 bit-stream compared with cascaded pixel-domain scheme.

As in the case of intra-MB conversion, Eq. 6 for unquantized coefficients is derived from the MPEG-2 quantization formula, Eq. 5, where $W_{(i,j)}^{non\text{-}intra}$ indicates $(i,j)^{\text{th}}$ element of the quantizer matrix for non-intra MB, and the newly quantized coefficients $\hat{B}_{(i,j)}/q'$ can be calculated by dividing $\hat{B}_{(i,j)}$ by q'.

3.2 DCT Conversion

The 8×8 DCT block conversion into four 4×4 approximate DCT blocks is extended from the pixel-domain extracting. In the pixel-domain, extracting of the 4×4 sub-block B_i from B is defined by Eq. 8, where B is an 8×8 motion compensated (MC) block or a intra block in the pixel-domain, and $L_i, R_i, i = 0,..,3$ is the displacement ma-

trices that perform vertical and horizontal filtering, respectively. The order of sub-block B_i is a raster scan order in B. The matrices L_i size of 4×8 and R_i size of 8×4 are defined in Eq. 9, where $I_{4\times4}$ is identity matrix of size 4. The pre-multiplication by L_i vertically extracts a sub-block size of 4×8 from B whereas, the post-multiplication by R_i horizontally extracts 4×4 block from the resultant block.

$$B_i = L_i \cdot B \cdot R_i (0 \le i \le 3) \tag{8}$$

$$L_0 = L_1 = \left(I_{4\times4} \quad 0\right)_{4\times8}, \; L_2 = L_3 = \left(0 \quad I_{4\times4}\right)_{4\times8},$$
$$R_0 = R_2 = \begin{pmatrix} I_{4\times4} \\ 0 \end{pmatrix}_{8\times4}, \; R_1 = R_3 = \begin{pmatrix} 0 \\ I_{4\times4} \end{pmatrix}_{8\times4} \tag{9}$$

The 4×4 DCT block can be calculated by performing DCT on the extracted sub-block, B_i of Eq. 8. Because of the distributive property of matrix multiplication with respect to the DCT, transformed matrix of B_i is given by $\hat{B}_i = \hat{L}_i \cdot \hat{B} \cdot \hat{R}_i$ as represented in Eq. 10, where $\hat{B}_i$, $\hat{L}_i$, $\hat{B}$, and $\hat{R}_i$ are the DCT representations of B_i, L_i, B and R_i respectively. Especially, $\hat{L}_i$ and $\hat{R}_i$ can be written in the form of $H_{4\times4} \cdot L_i \cdot H^T_{8\times8}$ and $H_{8\times8} \cdot R_i \cdot H^T_{4\times4}$ as described in Eq. 11, where $H_{4\times4}$ and $H_{8\times8}$ are the transform matrices for 4×4 and 8×8 DCT, respectively, and $H^T_{4\times4}$ indicates the transpose of $H_{4\times4}$. Consequently, $\hat{B}_i$ can be rewritten as the form of Eq. 11. Through this equation, we can simply change an 8×8 DCT block into four 4×4 DCT blocks, because the transform matrices, $H_{4\times4}$, $H^T_{4\times4}$, $H_{8\times8}$, $H^T_{8\times8}$, and filtering matrices L_i, $R_i (0 \le i \le 3)$ are already known matrices and we can make up the look-up table using them. However, $\hat{B}_i \; (0 \le i \le 3)$ is not identical with the 4×4 approximate DCT block, $\tilde{B}_i$ of H.264/AVC, because the transform matrices are different as mentioned in section 2. Accordingly, it is necessary to compensate the difference, and Eq. 12 shows our final formula. In Eq. 12, we can obtain $\tilde{B}_i$ by using $H'_{4\times4}$, which changes elements of $H_{4\times4}$, $b = \sqrt{1/2}\cos(\frac{\pi}{8}) \approx 0.6533$ and $c/b = 0.414$ to $b = \sqrt{2/5}$ and $c/b = 1/2$, instead of $H_{4\times4}$. The specific derivation processes are shown in Eq. 13 ~ Eq. 15. In Eq. 14, $H_{8\times8}{}^T H_{8\times8}$ and $H_{8\times8} H_{8\times8}{}^T$ are an identity matrix, respectively, due to the orthogonal property of the DCT transform matrix. Lastly, we can get the form of Eq. 15 by applying the associative property of matrix multiplication to Eq. 14. This Equation is equal to Eq. 12, because the matrix multiplication $H_{8\times8} \cdot B \cdot H^T_{8\times8}$ is identical to $\hat{B}$. As a consequence, we can exactly converts an 8×8 DCT block into four 4×4 approximate DCT blocks of H.264/AVC with the modified transform matrix $H'_{4\times4}$. As stated earlier, the product of matrices, $H'_{4\times4} \cdot L_i \cdot H^T_{8\times8}$ and $H_{8\times8} \cdot R_i \cdot H'^T_{4\times4}$ in Eq. 12 can be pre-computed and stored in the memory as a look-up table.

$$\begin{aligned} DCT(B_i) = DCT(L_i \cdot B \cdot R_i) &= DCT(L_i) \cdot DCT(B) \cdot DCT(R_i) \\ &= \hat{B}_i = \hat{L}_i \cdot \hat{B} \cdot \hat{R}_i, \quad (0 \le i \le 3) \end{aligned} \tag{10}$$

$$\hat{B}_i = (H_{4\times4} \cdot L_i \cdot H_{8\times8}^T) \cdot (\hat{B}) \cdot (H_{8\times8} \cdot R_i \cdot H_{4\times4}^T) \quad (0 \le i \le 3) \tag{11}$$

$$\tilde{B}_i = (H'_{4\times4} \cdot L_i \cdot H_{8\times8}^T) \cdot (\hat{B}) \cdot (H_{8\times8} \cdot R_i \cdot H'^T_{4\times4}) \quad (0 \le i \le 3) \tag{12}$$

$$\tilde{B}_i = H'_{4\times4} \cdot B_i \cdot H'^T_{4\times4} = H'_{4\times4} \cdot (L_i \cdot B \cdot R_i) \cdot H'^T_{4\times4} = (H'_{4\times4} \cdot L_i) \cdot B \cdot (R_i \cdot H'^T_{4\times4}) \tag{13}$$

$$= (H'_{4\times4} \cdot L_i) \cdot (H_{8\times8}^T \cdot H_{8\times8}) \cdot B \cdot (H_{8\times8}^T \cdot H_{8\times8}) \cdot (R_i \cdot H'^T_{4\times4}) \tag{14}$$

$$= (H'_{4\times4} \cdot L_i \cdot H_{8\times8}^T) \cdot (H_{8\times8} \cdot B \cdot H_{8\times8}^T) \cdot (H_{8\times8} \cdot R_i \cdot H'^T_{4\times4}) \tag{15}$$

4 Experimental Results

Throughout the various experiments we compare the performance of our proposed scheme with the cascaded re-encoding method of pixel-domain. In this paper, we call the cascaded re-encoding method "cascade" method. For simulation, we implemented DCT-based transcoding architecture and cascaded pixel-based transcoding architecture by modifying the MPEG-2 Test Model 5 (TM5) codec [12] and adopting the H.264 Joint Model 8 (JM8) encoder partially [13]. For convenient simulation, we fixed Qstep for all frames in a video sequence. In experiments, we tested three video sequences with the different motion characteristic: FOOTBALL (high motion degree), CARPHONE (medium degree), and CLAIRE (low degree). In the simulation, input video sequences of Quarter Common Intermediate Format (QCIF) were encoded by the MPEG-2 encoder at constant frame rate of 30, GOP size of 6, and I/P distance of 3, and then, the output bit-streams of the MPEG-2 encoder are transcoded to H.264 bit-streams.

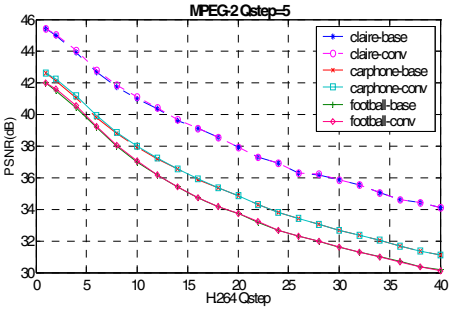


Fig. 1. The comparison of PSNR for test video sequences when Qstep is fixed to 5 in MPEG-2 encoding and the re-Qstep is changed from 1 to 40

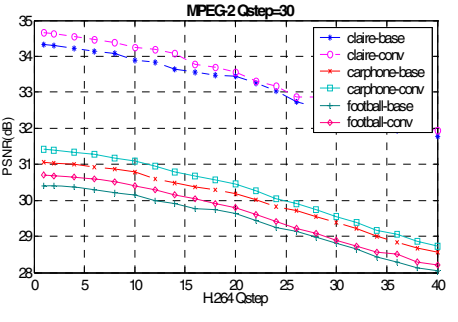


Fig. 2. The comparison of PSNR for test video sequences when Qstep is fixed to 30 in MPEG-2 encoding and the re-Qstep is changed from 1 to 40

Fig. 1 and Fig. 2 shows the peak signal-to-noise ratio (PSNR) comparison as changing the re-Qstep from 1 to 40 of the transcoder at fixed MPEG-2 Qstep 5 and 30, respectively. In the legend of the figures, '-base' refers to cascade method and '-conv' refers to our proposed scheme. In Fig. 1, we cannot make a discrimination between the

performance of the proposed scheme and the cascade method even though our method is numerically higher than the cascade method by 0.01 ~ 0.04 dB. Contrast to this result, Fig. 2 shows different results that PSNR of our method is higher than the cascade method by maximum 0.45 dB. This is due to our quantization scheme that has an effect of skipping MPEG-2 quantization as explained in section 3.1.

Fig. 3 and Fig. 4 show PSNR of the frames ranging from 1 to 30 for CARPHONE and CLAIRE sequences, respectively. With this figures, we can observe that, as a whole, the proposed scheme shows higher performance than the cascade method. Especially, for I frames, it is more superior to the cascade method. The reason is due to the magnitude of the Motion Compensated DCT (MC-DCT) coefficients. More specifically, because the magnitude of the intra-coded block coefficients is generally larger than the inter-coded block coefficients, the quantization errors of the intra-coded block are larger than the inter-coded block at high Qstep. Conversely, because our scheme has an effect of skipping the MPEG-2 quantization process, it can maintain the video quality.

Fig. 5 shows the reconstructed real images carried out the transcoding process at Qstep 20 for MPEG-2 encoding and at re-Qstep 6 for H.264/AVC transcoding. The cascade method images are arranged in the first row, while the proposed method images are in the second row. FOOTBALL, CARPHONE, and CLAIRE images are arranged from left to right. Let us look hard at the images and compare each pairs. First, for the FOOTBALL sequence, the second player's hip area of our image (b) is clearer than that of the cascade method (a). Second, for CARPHONE, we cannot discover notable difference. Third, for CLAIRE, the woman's silhouette and background of our image (f) are clearer without blurring than the cascade method (e).

Finally, we represent the computational complexity by comparing the number of multiplications and additions. We have implemented IQ, IDCT, DCT, and re-Q for the cascade method, and quantization conversion and DCT conversion for the proposed method, respectively. For the cascade method, we do not consider fast DCT/IDCT methods, but only consider the conventional DCT/IDCT methods. Fast DCT/IDCT methods can reduce computational complexity by re-using intermediately calculated values instead of several multiplications and additions for transform. Even though our proposed method also can be implemented by the same kind of fast

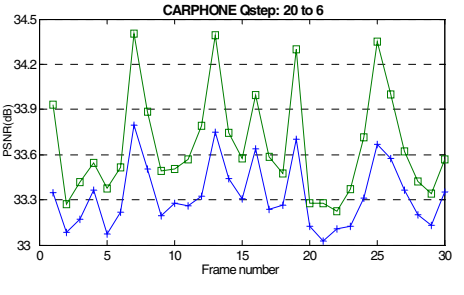


Fig. 3. The PSNR performance for the CARPHONE sequence ranging from frame 1 to frame 30, where Qstep is set to 20 for MPEG-2 encoding and re-Qstep is set to 6 for H.264/AVC

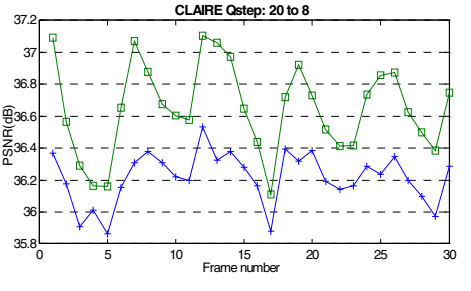


Fig. 4. The PSNR performance for the CLAIRE sequence ranging from frame 1 to frame 30, where Qstep is set to 20 for MPEG-2 encoding and re-Qstep is set to 8 for H.264/AVC

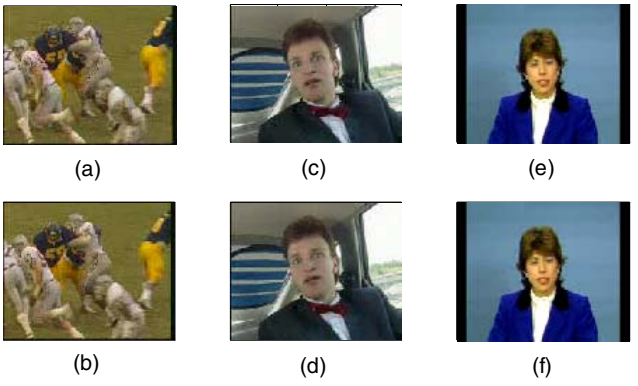


Fig. 5. The real images for the sample video sequences at Qstep=20 and re-Qstep=6: FOOTBALL (a)(b) CARPHONE (c)(d) CLAIRE (e)(f). The cascade method images are arranged in the first row, while the proposed method images are in the second row.

Table 1. The required operations to transcode a quantized 8×8 DCT block to newly quantized 4×4 approximate DCT blocks

Function	Cascade method (Inter or Intra block)	Proposed method (Inter)	Proposed method (Intra)
IQ/Q (or Qcon)	192M+128A+192S	64M+1D	128M+1D+64A+64S
DCT conversion	-	1,024M+896A	-
8×8 IDCT	1,024M+896A	-	1024M+896A
4×4 approximate DCT	256A+64S	-	256A+64S
Total operations	1,216M+1,280A+256S	1,088M+1D+896A	1,152M+1D+1,024A+128S

method, we do not consider the fast method in this paper. For intra-coded block, we use the cascaded re-encoding method in pixel domain to avoid too much computation complexity.

Table 1 shows the number of operations to transcode a quantized block of 8×8 size of MPEG-2 to newly quantized four 4×4 approximate DCT blocks of H.264/AVC, where "M" stands for multiplication operations, "A" for adds, "S" for Shift, and "D" for division. For a brief comparison of computational complexity, we ignore the number of addition, shift, and subtraction operations because the overhead of the multiplication and division operations are higher than any other operation. According to Table 1, the proposed scheme allows a computational complexity saving of about 11% for non-intra-coded blocks and about 5% for intra-coded blocks, compared with the cascade method.

5 Conclusions

In this paper, we proposed an efficient and exact conversion algorithm for the quantized 8×8 DCT block of MPEG-2 into newly quantized four 4×4 approximate DCT blocks of H.264/AVC to support DCT-domain transcoding. With the slight modification of quantization conversion, this method also can be applied to MPEG-1, MPEG-4 and H.263, etc. Extensive simulation results show that the PSNR of the proposed method outperforms the cascaded re-recoding method in the pixel domain by 0.1 ~ 0.5 dB and the computational complexity can be reduced by 5~11%. Our next subject is to find out an efficient half pixel conversion algorithm to compensate the difference of half pixel value between MPEG-2 and H.264/AVC and is to integrate the algorithm and our DCT conversion algorithm.

References

1. ISO/IEC 14496-10:2003, Coding of Audiovisual Objects –Part 10: Advanced Video coding. 2003 and ITU-T Recommendation H.264 :Advanced video coding for generic audiovisual services.
2. Vetro, A.; Christopoulos, C.; Sun, H, "Video Transcoding Architectures and Techniques: An Overview", IEEE Signal Processing Magazine, Vol.20, Issue2, pp.18-29, March 2003.
3. H. Sun, W. Kwok, and J. Zdepski, " architectures for MPEG compressed bitstream scaling", IEEE Trans. Circuits Syst. Video Technol., vol. 5, pp.191-199, April 1996.
4. N. Bjork and C. Christopoulos, "Trascoder architectures for video coding", IEEE Trans. Consumer Electron., vol.44, pp.88-98, February 1998.
5. J. Youn, M.T. Sun, and C.W. Lin, "Motion vector refinement for high performance transcoding", IEEE Trans. Multimedia, vol.1, pp.30-40, March 1999.
6. J.N. Hwang, T.D. Wu, and C.W. Lin, "Dynamic frame-skipping in video transcoding", In Proc. IEEE Workshop Multimedia signal processing, Redondo Beach, CA, pp.616-621, December1998.
7. Kalva, H.; Vetro, A.; Sun, H., "Performance Optimization of the MPEG-2 to MPEG-4 Video Transcoder", SPIE Conference on VLSI Circuits and Systems, Vol. 5117, pp.341-350, May 2003.
8. Haiyan Shu, "An Efficient Arbitrary Downsizing Algorithm for Video", Transcoding, IEEE Trans. Circuits Syst. Video Technol., vol.14, no.6, pp.887-891, June 2004.
9. Kwang-deok Seo, Jae-Kyoon Kim: "Fast motion vector re-estimation for transcoding MPEG-1 into MPEG-4 with lower spatial resolution in DCT-domain", Signal Processing: Image Communication, vol.19, pp.299-312, 2004.
10. Iain E.G. Richardson, H.264 and MPEG-4 video compression. Willey, 2003.
11. Henrique S. Malvar, "Low-Complexity Transform and Quantization in H.264/AVC", IEEE Trans. Circuits. Syst. Video Technol,. vol.13. no.7, pp.598-602, July 2003.
12. http://www.mpeg.org/MPEG/MSSG/tm5/
13. http://iphome.hhi.de/suehring/tml/download/

A Novel Watermarking Algorithm for Image Authentication: Robustness Against Common Attacks and JPEG2000 Compression

Marco Aguzzi, Maria Grazia Albanesi, and Marco Ferretti

Università degli Studi di Pavia, Pavia PV 27100, Italy
marco.aguzzi, mariagrazia.albanesi, marco.ferretti@unipv.it
http://orfeo.unipv.it

Abstract. This paper presents an authentication algorithm based on robust watermarking techniques. In particular, the proposed method is based on a *self - embedding* scheme that is able not only to authenticate noisy images, but also to recover areas modified by a software pirate. The attack method investigated are semantic (altering the meaning of what the image is about) tampering, Gaussian white noise superposition, and JPEG2000 compression. The results are checked against the TAF function, which measure the distance between the inserted and the extracted watermark, and compared to similar algorithms in literature.

1 Introduction

The paper is organized as follows: section 2 is about the Golay code [1], [2], [3], which serves as a foundation to the watermarking algorithm, section 3 is dedicated to introducing the watermarking algorithm, focusing on the wavelet transform, the embedding and the retrieval of the watermark; section 4 is about how the attacks to the watermarked image have been carried out, and section 5 is about the conclusion of the paper and the future works that we are going to develop.

2 The Golay Code

Our watermarking algorithm relies on Golay code. This theoretic foundation allows two main features: a) embedding the watermark and b) if an attack is performed on the watermarked images, reconstructing a new recovered image. This section serves as a theoretical introduction to the argument in order to support the next sections. The Golay code belongs to the family of the linear correcting codes. Let's assume that a message x of n digit is sent by a transmitter, while a message y is received by a receiver: generally we will assume $y = x + e$, where e is the error introduced either by a noisy channel or by a software attack. All equations written in this section are modulo 2, in order to use matrix notation and to adopt the convention $1 + 1 = 0$.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 559–567, 2005.

The *codeword* x does not contain only the information, but it is divided into k digits of actual information and $n-k$ check digits. The way the check digits are calculated depends on the particular code implementation, and it can be asserted by specifying the equation used for going from u (the actual data) to the codeword x or by constructing a k by n matrix in which the position i, j is 1 if the ith digit of u is used to form the jth digit of x. The matrix built in this way is called the *generator matrix* G and for this class of code it can always be constructed as

$$G = [I_k|P] \tag{1}$$

where I_k is the identity matrix of dimension k, P tells how the check digits have to be built, and $[\cdot|\cdot]$ denotes matrix concatenation. So building the codeword can be written as:

$$x = uG \tag{2}$$

On the other side of the communication channel, recalling that the decoder has received y instead of x a vector s, called the *syndrome*, is built using the equation

$$s = yH^T \tag{3}$$

where H is obtained as

$$H = [P^T|I_{n-k}] \tag{4}$$

The syndrome s is a $n-k$ element vector and it contains a 1 where the corresponding parity check has failed, 0 otherwise. If we want to use this kind of codes to correct the errors introduced by transmission or attacks, the information contained in the syndrome are not sufficient, because they signal only if an error is present and what the corresponding parity check failed is, but not where the actual error is. Recalling that $y = x + e$ and, relying on the properties of this class of codes, that $xH^T = 0$, Equation 3 can be rewritten as

$$s = eH^T \tag{5}$$

For each given syndrome the possible associated error patterns are 2^k: this means that even for low n and k it is not possible to store and look up all the error patterns associated with all the syndromes. So only a subset of all the error patterns will be actually correctable. In order to minimize the decoding error produced when the actual error that corresponds to the obtained syndrome differs from the correctable one, the choice of the error patterns eligible for correction points to the ones with the minimum weight, that is, that contain the least number of 1's. The table containing the syndromes and their selected error patterns is called *standard array*.

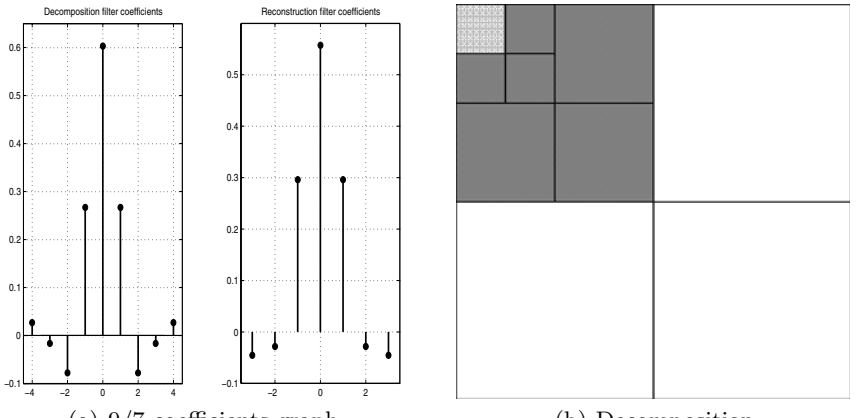
The Golay code is part of a subset of this class of codes: the study of the structure of G, H, and standard array is beyond the scope of this paper. What has to be known is that the code has $n = 23$ and $k = 12$ and is able to correct triple errors.

3 Description of the Novel Algorithm

3.1 Embedding the Watermark

We will focus in more detail in the embedding of the watermark because most of the transformations of this phase are used in the extracting process without modifications. The steps that have to be performed to insert the watermark are: a) wavelet decomposition, b) coefficients quantization, c) watermark calculation on baseband coefficients (see light gray area in Figure 1(b)), and d) watermark embedding in coefficients coming from the next two levels of the decomposition. (see dark gray area in Figure 1(b))

Wavelet Decomposition. The wavelet transform chosen for embedding the watermark uses the Daubechies 9,7 [4] filters. The properties of these filters [5] are: they are compactly and not infinitely supported, they have the shortest basis function for wavelets with 4 vanishing moments, and they form a nearly orthogonal basis. The filter coefficients belonging to the 9,7 filter are reported in Figure 1(a), using their traditional graphic representation. The wavelet decomposition is done over three levels, as depicted in Figure 1(b).



(a) 9/7 coefficients graph (b) Decomposition

Fig. 1. The coefficients used for wavelet decomposition and its subband representation

Coefficients Quantization. The coefficients produced by the wavelet transform are quantized accordingly to the Watson [6] step. Indeed, the thresholds found by Watson are strictly adapted to the wavelet transform and chosen to make the quantization noise perceptually invisible. The quantization matrix is calculated using the formula reported in [6].

Watermark Computation. The baseband coefficients are ported to a suitable data structure containing only their binary digits, scrambled according to a

random permutation and divided into k-element ($k = 12$) blocks. Each k-element block generates the bit string u that builds the watermark in an incremental fashion. The equation used is similar to (2), but here only the parity digits are produced; thus the actual equation used is

$$\hat{x} = uP \tag{6}$$

So $\hat{x}$ is the $n - k$ digit vector containing the check digits of the codeword x. The watermark is formed by successive concatenations of vector $\hat{x}$, which is generated for each bit plane of the baseband coefficient and for each k-element block.

Watermark Embedding. The embedding process does not wait for the whole watermark to be produced, but the insertion goes along as the $\hat{x}$ vectors are ready. The coefficients of these sub bandsare also converted in the "binary digits" structure, scrambled according to a random permutation, divided into $(n - k)$-element blocks ($n - k = 11$) and the least significant bit plane of these blocks is substituted with the corresponding piece of watermark, that is the vector $\hat{x}$.

3.2 Extracting the Watermark

The extraction of the watermark partly follows the same steps as the embedding. The wavelet transform is applied to the watermarked image, and the coefficients are quantized in the same way as when the watermark was being embedded. The quantized coefficients are grouped accordingly to the scheme in Figure 1(b), and on each of them a random permutation is performed, initializing the random seed in the same way as it was initialized when embedding the watermark; this to assure that the permutations in this step are carried out in the same way as the ones made in the embedding process. From the baseband coefficients the vector u is extracted and from the watermarked "embedding" coefficients the vector $\hat{x}$ is extracted. The codeword x is built by the concatenation of u and $\hat{x}$ which are, respectively, of k and $n - k$ elements. The syndrome s is calculated using Equation 3. Now if one ore more digits of s are different from 0, the codeword received has some errors: the corrected codeword is obtained subtracting modulo 2 from the mistaken codeword the value of the standard array indexed by the syndrome.

Image Reconstruction. From the corrected coefficients and the error mask built while performing the watermark extraction according to the calculated syndrome values, the coefficients of the reconstructed image, before applying the inverse wavelet transform, are calculated in this way:

$$u_r = y_c v + y\overline{v} \tag{7}$$

where u_r is the overall reconstructed coefficients, y_c the received coefficients after correction and v is the error mask. The result from Equation 7 is joined with unchanged (and uncorrectable) coefficients from the other bands and on them the inverse wavelet transform is performed in order to obtain the reconstructed image.

4 Robustness to Attacks

When a robust watermark scheme is adopted, a possible measure of how good the watermark is relies on the Tamper Assessment Function (TAF) [7] shown in Equation 8: given the embedded watermark w^M and the extracted watermark w^X, both NW bit long, it measures how much the two watermarks differ one from the other.

$$TAF(w^M, w^X) = \frac{\sum_{i=1}^{NW} w_i^M \oplus w_i^X}{NW} \tag{8}$$

This ratio floats between 0 and 1, and among these two a threshold th can be set in order to discriminate between tampered or not tampered images. In the next subsections various types of attack made to the image will be described, and it will be shown the effects that those attacks have on the TAF value.

As a quality measure also the Peak Signal to Noise Ratio (PSNR) will be considered. The formula used for calculating the PSNR is shown in Equation 9, given the reference signal s^r and the noisy signal s^n, both N^2 samples long. We consider a square image (N rows by N columns) and 255 as the maximum image sample value.

$$PSNR(s^r, s^n) = 10 \log_{10} \left(\frac{255^2}{\frac{1}{N^2} \sum_{i=1}^{N^2} (s_i^r - s_i^n)^2} \right) \tag{9}$$

It will be shown that as the PSNR increases, the TAF decreases consequently, even though the former measure seems to be less "sensitive" than the latter.

4.1 Semantic Tampering

With the term "semantic tampering" we address those kinds of attack aimed at corrupting the semantic meaning of the image. For example, Lena's nose can be substituted with an "ugly" one. Three examples of this kind will be shown: for each attack will we show: a) the original image, b) the watermarked and attacked image, and c) the reconstructed image. (In the case of Lena the original image is not shown because it is already well known.) In Table 1 we report, for each attack, the PSNR loss and the calculated TAF.

Table 1. Semantic attacks summary

	PSNR	PSNR loss	TAF
Bird without feather	37.0305	2.3971	0.0126
Bird with two feathers	39.4276	1.8787	0.0108
"Ugly" *Lena*	37.9413	1.2891	0.0051

4.2 Adding White Noise

For the second attack, we superimposed a white noise, generated by a Gaussian distribution with mean value $\bar{x} = 0$ and varying σ, over the watermarked image.

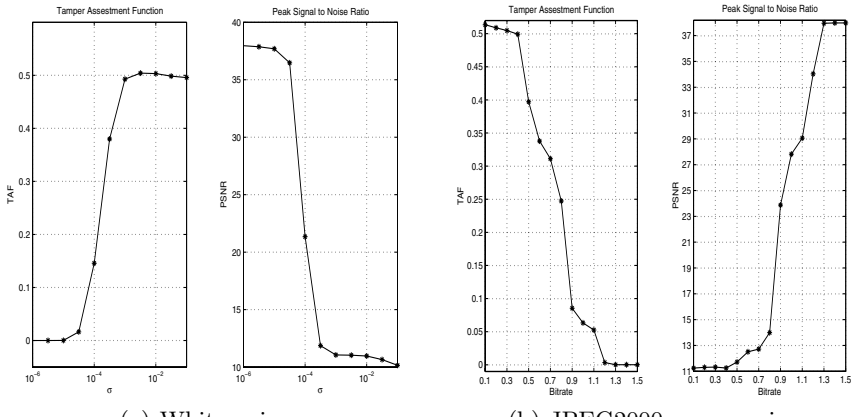
The alteration is spread all over the image, and for increasing σ the TAF increases and the PSNR decreases. σ ranges from $10^{-}1$ to $10^{-}5$, and the curves, as shown in Figure 2(a), highlight the breakpoint around $\sigma = 10^{-}4$, where the quality of the reconstruction becomes unacceptable.

4.3 JPEG2000 Compression

The third attack is related to JPEG2000 compression. The original image is watermarked and then compressed with a bit rate varying from 0.1 bpp (bit per pixel) to 1.5 bpp. As the graphs plotted in Figure 2(b) show, the compression becomes no longer acceptable when the compression rate goes under 1 bpp.

A second piece of information that is visible in Figure 2(b) is in correspondence of the final plateau in the two graphs: they show when the PSNR and the TAF become no longer "sensible" to a relaxation in the compression rate: the TAF remains equal to zero (this means that the watermark is perfectly extracted) after 1.2103 bpp, the PSNR becomes stable after 1.3 bpp. The stable value that the PSNR assumes is slightly different from the original obtained value of 37.94 dB, this is because the compression rate specified as a parameter from the JPEG2000 encoder command line is not strictly equal to the actual gain obtained during compression, so the reconstructed image seems even better than the watermarked one.

Compared to the values reported in work [7], the TAF values we obtained are much better. In Table 2 are reported TAF values obtained with JPEG and



(a) White noise (b) JPEG2000 compression

Fig. 2. Tamper assestment function and Peak Signal to Noise Ratio for gaussian white noise superimposal with a σ varying from 10^{-6} to 10^{-1}. and for JPEG2000 compression varying from 0.1 to 1.5 bits per pixel

Table 2. TAF values comparison with results in [7]. Conversion between bpp and Compression Ratio is made speculating that gray level images in [7] are taken at 8 bpp

bpp	Compression Ratio	TAF [7]	our TAF
4	2	0.0615	0
2.67	3	0.0732	0
2.29	3.5	0.1357	0
1.78	4.5	0.1455	0
1.6	5	0.1729	0
1.33	6	0.25	0
1.14	7	0.3027	0.0544
0.94	8.5	0.4004	0.0891
0.8	10	0.4229	0.2472
0.7	11.43	n.a.	0.3111
0.6	13.33	n.a.	0.3377
0.5	16	n.a.	0.3970
0.4	20	n.a.	0.4991
0.3	26.66	n.a.	0.5046
0.2	40	n.a.	0.5088
0.1	80	n.a.	0.5134

JPEG2000 compression. Bit rate ranges from 1.333 bpp to 0.8 bpp. A wider range would not have been possible because a) for compressions lighter than 1.3 bpp our algorithm returns always a TAF value equal to 0, and b) the results reported in [7] go no further than 0.8 bpp, while ours do.

5 Conclusions and Future Work

This work has shown a *self-embedding* robust watermark inserted into images based on the Golay code using wavelet transform. The method has been tested over various kinds of attack, and the most interesting thing is that the performances over JPEG2000 compression are quite good, allowing compression toward at almost 1 bit per pixel while preserving watermark extraction. Other kinds of attack, such as semantic ones, are well localized by the algorithm so that substitution with lower resolution data is well localized as well. As future work, we are planning to use this approach in order to embed watermarking and public / private key criptography in authentication of bio-metrical data.

References

1. Benedetto, S., Biglieri, E., Castellani, V.: Digital Transmission Theory. Prentice - Hall, Inc. (1987)
2. Gallagher, R.G.: Information theory and reliable communications. John Wiley & Sons (1968)

3. Golay, M.J.E.: Notes on digital coding. Proceedings of IEEE **37** (1949) 637
4. Daubechies, I.: Ten lectures on wavelet. Society for Industrial and Applied Mathematics (1992)
5. Unser, M., Blu, T.: Mathematical properties of the jpeg2000 wavelet filters. IEEE Transaction on Image Processing **12** (2003) 1080–1090
6. Watson, A.B., Yang, G.Y., Solomon, J.A., Villasenor, J.: Visibility of wavelet quantization noise. IEEE Transaction on Image Processing **6** (1997) 1164–1175
7. Kundur, D., Hatzinakos, D.: Digital watermarking for telltale tamper proofing and authentication. Prooceedings of IEEE **87** (1999) 1167–1180 Invited paper.

(a) Original *Bird*

(b) Watermarked and tampered *Bird*

(c) Reconstructed *Bird*

Fig. 3. *Bird* without feather

(a) Watermarked and tampered *Bird* (b) Reconstructed *Bird*

Fig. 4. *Bird* with two feathers

(a) Watermarked and tampered *Lena* (b) Reconstructed *Lena*

Fig. 5. "Ugly" *Lena*: the nose is heavily tampered

A Video Watermarking Procedure Based on XML Documents

Franco Frattolillo and Salvatore D'Onofrio

Research Centre on Software Technology,
Department of Engineering, University of Sannio, Italy
frattolillo@unisannio.it

Abstract. This paper presents a watermarking procedure for MPEG-2 videos based on the use of XML documents. The procedure enables the copyright owner to insert a distinct watermark identifying the buyer within the distributed videos. To increase the security level of the procedure, the watermark is repeatedly embedded into some selected I-frames in the DCT domain at different frequencies and by exploiting both block classification techniques and perceptual analysis. The embedded watermark is then extracted from a video according to the information contained in a protected XML document associated to the video.

1 Introduction and Motivations

Digital watermarking [1] can be considered a main technology to implement the copyright protection of digital contents distributed on the Internet. To this end, many recent watermarking procedures achieve high levels of security and robustness by adopting "readable" watermarking schemes based on "blind" and not publicly available decoders as well as letting the inserted watermarks depend on the host signals [2]. However, the "nonblind" watermarking schemes are typically considered more robust then the blind ones [3,4]. Unfortunately, differently from the blind ones, the nonblind schemes need the original contents to run the watermark detection or extraction algorithms. This is considered a relevant drawback particularly for the procedures that aim at being adopted in a web context, because such procedures force the distinct web entities involved in "identification and arbitration" protocols [4] to exchange and store the original unprotected, large size digital contents, thus risking "colluding" behaviors [2,3,4,5].

This paper presents a watermarking procedure for MPEG-2 compressed videos based on the use of XML documents. The procedure enables the copyright owner to insert a distinct code identifying the buyer within each copy of the distributed videos. Furthermore, to increase the security level of the procedure, the watermark is repeatedly embedded into some selected I-frames of a video in the DCT domain at different frequencies and by exploiting both block classification techniques and perceptual analysis. The embedded watermark is then extracted from a video according to the information contained in a protected XML document associated to the video. Thus, the usual security and robustness

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 568–575, 2005.

levels characterizing the nonblind watermarking schemes can be achieved without requiring the original unprotected, large size videos to be exchanged in the Internet whenever the watermark extraction has to be performed. Furthermore, using the XML technology makes it also easier to automate the document access in a web context, since XML is a technology well supported by the Java world, and document parsers, such as SAX and DOM parsers, are freely available.

The paper is organized as the follows. Section 2 describes the proposed watermarking procedure. Section 3 describes the watermark extraction procedure. Section 4 reports on some preliminary experimental results. Section 5 reports conclusion remarks.

2 The Watermarking Procedure

The watermarking procedure makes it possible to insert into an MPEG-2 video a binary code represented by a sequence of bits $\mu \in \{0,1\}$ and able to unambiguously identify a user. The sequence μ, whose length is denoted as n_μ, is repeatedly embedded into each of some selected I-frames of the video, denoted as $\epsilon_1, \epsilon_2 \ldots \epsilon_r$, in the DCT domain at different frequencies, denoted as $\gamma_1, \gamma_2 \ldots \gamma_f$. In particular, since the coefficients in each 8×8 DCT block have a frequency value associated with them, a γ value identifies an entry in such blocks, and so, it can range from 1 to $8^2=64$. Furthermore, to increase the security level of the procedure, the watermark insertion is assumed to be usually carried out at low, middle and high frequencies chosen on the basis of the video to watermark.

In principle, all the DCT coefficients of an I-frame could be modified by a value representing a watermark information. However, in the proposed procedure, the "perceptual capacity" of the coefficients belonging to the luminance DCT blocks of the selected I-frames is preliminarily estimated by exploiting both block classification techniques and perceptual analysis. In fact, the block classification techniques [6,7] are applied to indicate the bests DCT coefficients of the selected I-frames that can be altered without reducing the visual quality. They classify each luminance DCT block with respect to its energy distribution. The result of this procedure is a first selection of DCT coefficients whose modification has a minimal or no impact to the perceptual quality of the selected I-frames.

The perceptual analysis is then applied to calculate the "just noticeable difference" (jnd) values for the DCT coefficients [8,9,10]. Such values are the thresholds beyond which any changes to the respective coefficient will most likely be visible. Therefore, let $X^{\epsilon}_{b_m}(\gamma)$ denote the DCT coefficient at the frequency γ in the block b_m of the I-frame ϵ, and let $JND^{\epsilon}_{b_m}(\gamma)$ denote the jnd value calculated for the $X^{\epsilon}_{b_m}(\gamma)$ coefficient. $JND^{\epsilon}_{b_m}(\gamma)$ can be approximated by the following expression:

$$JND^{\epsilon}_{b_m}(\gamma) \approx \max\left\{C^{\epsilon}_{b_m}(\gamma), \left|C^{\epsilon}_{b_m}(\gamma)\right| E^{\epsilon}_{b_m}(\gamma)^g\right\} \tag{1}$$

where $C^{\epsilon}_{b_m}(\gamma)$ represents the perceptual threshold of the contrast masking and is expressed as:

$$C^{\epsilon}_{b_m}(\gamma) = \max\left\{t^{\epsilon}_{b_m}(\gamma), \left|X^{\epsilon}_{b_m}(\gamma)\right|^h t^{\epsilon}_{b_m}(\gamma)^{1-h}\right\} \tag{2}$$

$E^{\epsilon}_{b_m}(\gamma)$ is the entropy value calculated over the eight neighbors of the $X^{\epsilon}_{b_m}(\gamma)$ coefficient [8,9] and can be approximated by the following expression:

$$E^{\epsilon}_{b_m}(\gamma) \approx X^{\epsilon}_{b_m}(\gamma) - u^{\epsilon}_{b_m}(\gamma)q(\gamma) \tag{3}$$

In (1) g is assumed equal to 0.5, while in (2) h is assumed equal to 0.7 and $t^{\epsilon}_{b_m}(\gamma)$ is equal to $t(\gamma)(X^{\epsilon}_{b_m}(1)/X(1))$, where $X(1)$ is a DC coefficient corresponding to the mean luminance of the display, while $X^{\epsilon}_{b_m}(1)$ is the DC coefficient of the block b_m of the I-frame ϵ. In fact, $t(\gamma)$ can be approximated by the value $q(\gamma)/2$, where $q(\gamma)$ represents the coefficient of the quantization matrix corresponding to the frequency γ [9]. Finally, in (3) $u^{\epsilon}_{b_m}(\gamma)$ is equal to $round(X^{\epsilon}_{b_m}(\gamma)/q(\gamma))$.

The insertion procedure at the frequency γ of a selected I-frame assumes that each bit of the sequence μ is inserted into the I-frame by altering a pair of DCT coefficients associated to the frequency γ and chosen among the ones previously selected by applying the block classification techniques and perceptual analysis. In particular, the "choice rule" states that two DCT coefficients of a selected I-frame are allowed to belong to a same pair only if they have a similar value. Furthermore, to insert the bits of μ into an I-frame ϵ, the "encoding function" K has to be defined. K defines an encoding rule by which the bits 0 and 1 are translated to the symbols of the alphabet composed by $\{\nearrow, \searrow\}$, respectively called the *up* symbol and the *down* symbol. Thus, a sequence $\mu \in \{0,1\}$ is translated to a corresponding sequence $\sigma \in \{\nearrow, \searrow\}$ depending on the function K. For example, the sequence $\{01101\ldots\}$ is translated to the sequence $\{\nearrow\searrow\searrow\nearrow\searrow\ldots\}$, if K associates the *up* symbol to 0 and the *down* symbol to 1.

Let μ be a user sequence, and let σ be the sequence obtained by applying a K function. Let $\gamma_i, \forall i = 1 \ldots f$, be the insertion frequencies, and let $\epsilon_p, \forall p = 1 \ldots r$, be the selected I-frames of the video. Let $W^{\epsilon_p}_{b_m}(\gamma_i)$ denote the watermarked DCT coefficient at the frequency γ_i in the block b_m of the I-frame ϵ_p. A symbol of σ is inserted into a pair of DCT coefficients belonging to the blocks b_m and b_n, at the frequency γ_i of the I-frame ϵ_p, by the following expressions:

$$\begin{cases} W^{\epsilon_p}_{b_m}(\gamma_i) = X^{\epsilon_p}_{b_m}(\gamma_i) - JND^{\epsilon_p}_{b_m}(\gamma_i) \\ W^{\epsilon_p}_{b_n}(\gamma_i) = X^{\epsilon_p}_{b_n}(\gamma_i) + JND^{\epsilon_p}_{b_n}(\gamma_i) \end{cases} \text{to insert } \nearrow$$

$$\begin{cases} W^{\epsilon_p}_{b_m}(\gamma_i) = X^{\epsilon_p}_{b_m}(\gamma_i) + JND^{\epsilon_p}_{b_m}(\gamma_i) \\ W^{\epsilon_p}_{b_n}(\gamma_i) = X^{\epsilon_p}_{b_n}(\gamma_i) - JND^{\epsilon_p}_{b_n}(\gamma_i) \end{cases} \text{to insert } \searrow$$

$$\forall i = 1 \ldots f \quad \text{and} \quad \forall p = 1 \ldots r$$

In fact, as specified above, since the choice rule requires that $X^{\epsilon_p}_{b_m}(\gamma_i) \approx X^{\epsilon_p}_{b_n}(\gamma_i)$, $\forall i = 1 \ldots f$ and $\forall p = 1 \ldots r$, the insertion process ends up maximizing the difference existing between the coefficients of the pair according to the direction specified by the insertion symbol and by an amount that cannot compromise the final visual quality of the video. Consequently, the insertion process should be carried out according to the following rules:

1. The I-frames to watermark should qualify significant scenes of the video, and more than three consecutive I-frames should belong to each selected scene.

2. The insertion frequencies should be evenly distributed among the low, middle and high frequencies, and should be chosen so that possible attacks characterized by a filtering behavior on the video frames would end up reducing the final video quality. This can be achieved by selecting the frequencies characterized by high spectrum values, which, if filtered, can impair the video.
3. At each insertion frequency and for each selected I-frame, the pairs of the selected DCT coefficients should belong to spatial regions that cannot be cropped without impairing the video.

Once the symbols of the sequence σ have been inserted into the selected I-frames at the chosen frequencies, in order to increase the security level of the watermarking procedure against "collusion" attacks [3,5], it is necessary to hide the modifications made to the DCT coefficients of the I-frames. In fact, let $\gamma_1, \gamma_2 \ldots \gamma_f$ be the insertion frequencies chosen for the video, and let $\Sigma^{\epsilon_p}(\gamma_i)$ denote the sequences of the pairs of DCT entries $(b_m^{\epsilon_p}(\gamma_i), b_n^{\epsilon_p}(\gamma_i))$ that have been involved in the watermarking process for a given frequency γ_i, $\forall i = 1 \ldots f$ and for a given I-frame ϵ_p, $\forall p = 1 \ldots r$. It is worth noting that both the set of the frequencies γ_i and the sets $\Sigma^{\epsilon_p}(\gamma_i)$ are always the same for all the copies of a given video to protect. Consequently, the DCT coefficients modified at the different insertion frequencies in the selected I-frames remain the same for all the copies of the video. Therefore, in order to prevent malicious users from individuating the DCT coefficients modified by the insertion process, the procedure adds the *jnd* values modulated by a binary pseudo-noise sequence $\rho \in \{-1, 1\}$ to all the unmodified DCT coefficients of the selected I-frames of the watermarked video:

$$X_{b_k}^{\epsilon_p}(\gamma_i) = X_{b_k}^{\epsilon_p}(\gamma_i) + \alpha_k \rho_k JND_{b_k}^{\epsilon_p}(\gamma_i),$$
$$(\forall p = 1 \ldots r) \text{ and } ((i \neq 1 \ldots f) \text{ or } (i = 1 \ldots f \text{ and } b_k \notin \Sigma^{\epsilon_p}(\gamma_i)))$$

where $0 < \alpha_k < 0.5$ is a randomly varied amplitude factor.

2.1 The XML Documents

The capability of both repeatedly embedding a user code at different frequencies and hiding the watermarked DCT coefficients in each of the selected I-frames of the video can make the proposed procedure almost secure against the most common filtering, corrupting, removal, averaging and collusion attacks. However, the characteristics of the insertion process could make the procedure vulnerable to geometric attacks. Therefore, to increase the robustness level of the procedure against such attacks, the attacked videos should be geometrically re-synchronized before carrying out the watermark extraction.

To detect the most common geometric distortions applied to a watermarked video without having to use complex re-synchronization techniques, the proposed procedure makes use of some information about the video, which is assumed to be stored in a protected XML document associated to it. This allows the information to be stored in both textual and quantitative form. The textual information can individuate and describe the scenes which the I-frames belong to, and some evident and significant "feature points" and boundary segments of the selected

I-frames. The quantitative information can provide the original dimensions of the video, the coordinates of the feature points and selected boundaries, some Fourier descriptors and statistical moments of K-point digital boundaries, as well as the eigenvectors and eigenvalues of some well-defined regions of the selected I-frames. Thus, inverse geometric transformations can be performed on the attacked video in order to restore it before the watermark extraction [1,7].

Therefore, the XML document associated to each video to protect has to include: (1) the insertion frequencies $\gamma_1, \gamma_2 \ldots \gamma_f$; (2) the selected I-frames $\epsilon_1, \epsilon_2 \ldots \epsilon_r$, also specified in terms of significant scenes which they refer to; (3) the sets $\Sigma^{\epsilon_p}(\gamma_i)$, $\forall i = 1 \ldots f$ and $\forall p = 1 \ldots r$; (4) the definition of the encoding function K; (5) some further information about the original video, which can synthetically characterize the video and can be exploited to individuate possible geometric modifications performed on it.

3 The Watermark Extraction

The first operation to perform before carrying out the watermark extraction from a protected video is its geometric re-synchronization. Therefore, let Figure 1(a) be an I-frame of the "Cactus" video. Let Figures 1(b) and 1(d) be respectively the corresponding watermarked and attacked version. The geometric re-synchronization can be carried out by exploiting the information stored in the XML document associated to the "Cactus" video in order to build a reference picture (Figure 1(e)) whose dimensions coincide with the ones of the original I-frame. Then, the feature points connected by segments and specified by the XML document are to be reported on this picture. These points have been originally determined on the watermarked I-frame (Figure 1(c)) and are: the two upper edges of the glass, whose coordinates are (x_1, y_1) and (x_2, y_2); the upper left edge of the comb, specified by (x_3, y_3); the tip of the nose represented on the glass, specified by (x_4, y_4). The coordinates are referred to the X and Y axis, and the dimensions of the original I-frame are respectively d_x and d_y.

The successive operation consists in reporting the feature points connected by segments on the attacked version of the I-frame (Figure 1(d)). In particular, the attacked I-frame is a scaled and 45° rotated version of the watermarked I-frame (Figure 1(b)). To this end, it is worth noting that the feature points can be reported on the I-frame solely starting from the textual description provided by the XML document associated to the video (Figures 1(d) and 1(f)). This entails a natural approximation in individuating the feature points on the attacked I-frame, which can determine errors in the geometric re-synchronization process. However, the preliminary tests, conducted also on other videos, have shown that the proposed procedure is robust with respect to such approximations. In fact, the procedure has been able to ensure a correct watermark extraction provided that the rotation degrees and scale factors are determined with approximations in the range about ±6%. However, these limits have never been exceeded in the conducted, practical tests. To this end, Figure 2 shows the result of the worst re-synchronization performed on the scaled and 45° rotated version of

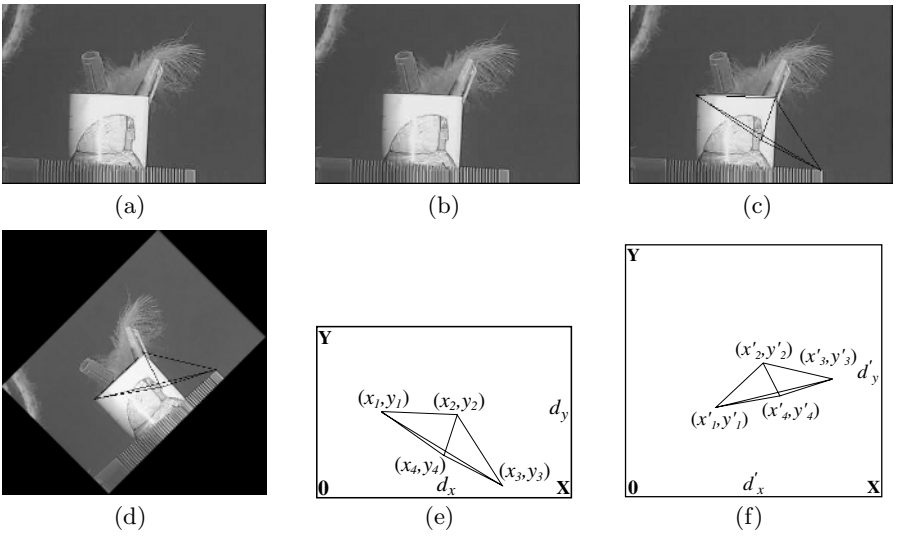


Fig. 1. The process of geometric re-synchronization of a "Cactus" I-frame

Fig. 2. A "Cactus" watermarked I-frame and its re-synchronized version

the watermarked I-frame, which has not anyway prevented a correct watermark extraction, as reported in Section 4. In fact, the imperfect re-building of the I-frame essentially affects the outer regions of the image, i.e. the regions that do not, and should not, host watermark information.

After the geometric re-synchronization, the watermark extraction can be carried out. In particular, for each insertion frequency γ_i and for each selected I-frames ϵ_p, the pairs of coefficients specified by the DCT entries $(b_m^{\epsilon_p}(\gamma_i), b_n^{\epsilon_p}(\gamma_i))$ belonging to the set $\Sigma^{\epsilon_p}(\gamma_i)$, with $i = 1 \ldots f$ and $p = 1 \ldots r$, have to be examined. Therefore, let $\hat{W}_{b_m}^{\epsilon_p}(\gamma_i)$ and $\hat{W}_{b_n}^{\epsilon_p}(\gamma_i)$ be the two coefficients of a pair belonging to $\Sigma^{\epsilon_p}(\gamma_i)$. To extract the watermark symbol they host, the following expression has to be calculated:

$$\begin{cases} \hat{W}_{b_m}^{\epsilon_p}(\gamma_i) - \hat{W}_{b_n}^{\epsilon_p}(\gamma_i) > 0 \implies \searrow \text{ is extracted} \\ \hat{W}_{b_m}^{\epsilon_p}(\gamma_i) - \hat{W}_{b_n}^{\epsilon_p}(\gamma_i) < 0 \implies \nearrow \text{ is extracted} \end{cases}$$

Then, the extracted symbol is translated to a bit depending on the K function.

After having completed the watermark extraction process, $f \cdot r$ user sequences $\hat{\mu_{i,p}}$ result in being re-built, one for each different insertion frequency γ_i and

Table A

Attack	*f*	*r*	*ber* (%)
Add Noise	3	9	2.03
Add Noise	6	12	1.97
Add Noise	9	15	1.85
Sharpening	3	9	1.64
Sharpening	6	12	1.56
Sharpening	9	15	1.48
Median	3	9	1.89
Median	6	12	1.62
Median	9	15	1.53
Rotating 45°	3	9	3.12
Rotating 45°	6	12	2.98
Rotating 45°	9	15	2.61

Table B

Attack	*f*	*r*	*k*	*ber* (%)	
Averaging	3	9	5	5.21	*Linear collusion attacks*
Averaging	6	12	10	6.05	
Averaging	9	15	15	6.93	
Minimum	3	9	5	3.46	*Non linear collusion attacks*
Minimum	6	12	10	3.91	
Minimum	9	15	15	4.38	
Median	3	9	5	4.16	
Median	6	12	10	4.93	
Median	9	15	15	5.67	
Maximum	3	9	5	3.52	
Maximum	6	12	10	3.99	
Maximum	9	15	15	4.37	

Fig. 3. The results of some preliminary tests

I-frame ϵ_p. Therefore, let $\mu(j)$ denote the j-th bit in the user sequence μ. $\mu(j)$ can be derived from the sequences $\mu_{i,p}^{\hat{}}$ by the following expression:

$$\mu(j) \equiv 1 \iff \frac{\sum_{i=1}^{f} \sum_{p=1}^{r} \mu_{i,p}^{\hat{}}(j)}{f \cdot r} > 0.5, \quad \forall j = 1 \ldots n_\mu \tag{4}$$

4 Experimental Results

The robustness and security of the proposed procedure have been assessed by measuring the bit error rate (*ber*) under a different number of the insertion frequencies f and of the watermarked I-frames r. The *ber* is expressed in percentage and is calculated as $(\sum_{i=1}^{f} \sum_{p=1}^{r} ber_{i,p})/f \cdot r$, where $ber_{i,p}$ is the number of bit errors reported in the watermark extraction carried out at the frequency γ_i from the I-frame ϵ_p. The user sequence μ is assumed 128 bit long, while the video used to perform the tests is the "Cactus" video and is coded at 30 fps, with a resolution of 328×224 pixels and a duration about 120 seconds. However, further tests have been conducted on other videos freely available on the Internet, but the obtained results have not been reported for the sake of brevity.

Table A in Figure 3 summarizes the results obtained under "simple" or "detection-disabling" attacks. In fact, the "simple" attacks attempt to impair the embedded watermark by manipulations of the whole watermarked video, such as filtering or compression manipulations or addition of noise. The "detection-disabling" attacks attempt to break the correlation and to make the recovery of the watermark impossible or infeasible, mostly by geometric distortion like rotation. Table B in Figure 3 shows the results obtained under some "linear" and "non linear" collusion attacks, whose correct definitions are reported in [5].In particular, k denotes the number of the colluding videos, while the anticollusion codes used in the tests have been generated according to what reported in [11].

The results reported in Figure 3 show that the proposed procedure can achieve a good performance against attacks that are considered able to prove a high level of robustness and security. In particular, the *ber* values are always low,

and the user sequence μ has been always correctly re-built from the sequences extracted for each test by applying the expression (4). In fact, the redundancy assured by the insertion process enables the procedure to behave as other well known, robust and secure watermarking procedures. This also because the procedure allows for choosing the insertion frequencies as well as the regions of the selected I-frames where to embed the watermark information. Moreover, in the conducted tests, the values of f and r are rather small, but they can be also increased in order to improve the security and robustness levels of the procedure without compromising the final visual quality of the video.

5 Conclusions

In this paper a watermarking procedure for the copyright protection of MPEG-2 videos is presented. The procedure directly acts on compressed videos and exploits protected XML documents to store information needed to the watermark extraction. The redundancy assured by the insertion process at different frequencies and for each of the selected I-frames enables the procedure to achieve a good performance against the most common and dangerous attacks. In addition, the procedure can also improve its security and robustness levels by increasing the number of the insertion frequencies and of the watermarked I-frames.

References

1. Cox, I., Bloom, J., Miller, M.: Digital Watermarking: Principles & Practice. Morgan Kaufman (2001)
2. Barni, M., Bartolini, F.: Data hiding for fighting piracy. IEEE Signal Processing Magazine **21** (2004) 28–39
3. Trappe, W., Wu, M., et al.: Anti-collusion fingerprinting for multimedia. IEEE Trans. on Signal Processing **41** (2003) 1069–1087
4. Lei, C.L., Yu, P.L., et al.: An efficient and anonymous buyer-seller watermarking protocol. IEEE Trans. on Image Proces. **13** (2004) 1618–1626
5. Wu, M., Trappe, W., et al.: Collusion-resistant fingerprinting for multimedia. IEEE Signal Processing Magazine **21** (2004) 15–27
6. Chung, T.Y., Hong, M.S., et al.: Digital watermarking for copyright protection of MPEG2 compressed video. IEEE Trans. on Consumer Electr. **44** (1998) 895–901
7. Wang, Y., Ostermann, J., Zhang, Y.: Video Processing and Communications. Prentice Hall (2002)
8. Kim, S.W., Suthaharan, S.: An entropy masking model for multimedia content watermarking. In: Procs of the 37th Hawaii Intl Conference on System Sciences, IEEE Computer Society (2004) 70182.2
9. Watson, A.B.: DCT quantization matrices visually optimized for individual images. In Allebach, J.P., Rogowitz, B.E., eds.: Human Vision, Visual Processing and Digital Display IV. Volume 1913 of SPIE Procs., S. Jose, CA, USA (1993) 202–216
10. Wolfgang, R.B., et al.: Perceptual watermarks for digital images and video. Procs of the IEEE **87** (1999) 1108–1126
11. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Trans. on Infor. Theory **44** (1998) 1897–1905

Half-Pixel Correction for MPEG-2/H.264 Transcoding*

Soon-young Kwon, Joo-kyong Lee, and Ki-dong Chung

Dept. Of Computer Engineering, Pusan National University, Busan, Korea
{ksy2020, jklee, kdchung}@pusan.ac.kr

Abstract. To improve video quality and coding efficiency, H.264/AVC [1] has adopted several newer coding tools such as a 4×4 integer DCT and a method for calculating half pixels than the previous standards. However, these tools require additional work to transcode video content for pre-coded in the previous standards to H.264/AVC in the DCT domain. In this paper, we propose, as far as we know, the first half-pixel correction method for MPEG-2 to H.264 transcoding in the DCT domain. In the proposed method, an MPEG-2 block is added to the correction block obtained from the difference between half-pixel values of the two standards using a DCT reference frame. Experimental results show that the proposed method achieves better quality than the pixel based transcoding method.

1 Introduction

H.264/AVC, the latest international video coding standard, has been approved by ISO/IEC and by ITU-T. According to [2], H.264 has a higher coding efficiency than any other previous video coding standard such as MPEG-2, H.263, and MPEG-4. Accordingly, it is expected to be in wide use and to replace other standards in the near future. This means that an efficient transcoding method that transcodes the currently widely available video coding standards into the H.264 compressed data format is needed [3]. The existent transcoders can be classified into two types of architectures, PDT (Open-loop Pixel-Domain Transcoder) and DDT (DCT-Domain Transcoder) [4]. Recent researches have focused on DDT approaches to improve computational complexity and to avoid the DCT and IDCT mismatch problem of PDT. However, so far, only PDT architecture has been used to transcode previous standards to H.264. That is, it is difficult to use common tools in transcoding because of the new tools in H.264. Specifically, H.264 uses a 6-tap filter to obtain half-pixel values and a 4×4 integer DCT instead of a 2-tap filter and an 8×8 float DCT. In this paper, we propose an efficient half-pixel correction scheme to support DCT-domain transcoding architecture from MPEG-2 to H.264. A correction scheme is essential for DCT-domain transcoding, because most video coding standards use half-pixel motion vector accuracy, but the half-pixel calculation method used in H.264 differs

* The work was supported by Pusan National University Research Grant.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 576 – 583, 2005.

from those of other standards. Our proposed scheme consists of two parts: CHPD (Correction for Half-Pixel Difference) and DCT_Conv (DCT Conversion). First, in CHPD, the MPEG-2 block is added to the correction block obtained by calculating the difference between the half-pixel values from two standards using a DCT reference frame. Second, the resultant block is passed into DCT_Conv to convert DCT transform between two standards. Both of the phases use matrices stored in memory to reduce computational complexity. DCT-domain H.264 transcoder requires our proposed method. And the proposed method supports previous transcoding technologies in the DCT-domain, such as frame rate control, bitrates reduction and scalability coding.

This paper is organized as follows. In Section 2 we describe the half-pixel operations of previous video standards and H.264. In Section 3 we present the proposed method and experimental results in Section 4. We close the paper with concluding remarks in Section 5.

2 Half-Pixel in Video Coding Standards

MPEG-2, H.263 and MPEG-4 use the same numerical formulas to calculate half-pixel values, however H.264 uses different formulas. The DDT to H.264, that reuses the half-pixel motion accuracy of the input video data, must perform the half-pixel correction. In this paper, we exploit three types of half-pixel MVs (Motion Vectors). They are, half-pixel horizontally, half-pixel vertically, and half-pixel bi-directionally, represented by H-halfpixel, V-halfpixel or HV-halfpixel, respectively. For example, the notations MV (1, 1.5), MV (1.5, 1), MV (1.5, 1.5) would correspond to each of them, respectively.

In MPEG-2, H.263 and MPEG-4, half-pixel values are the average of two or four neighboring pixels [5][6]. Fig. 1 illustrates the sample interpolation for samples *b*, *h* and *j*.

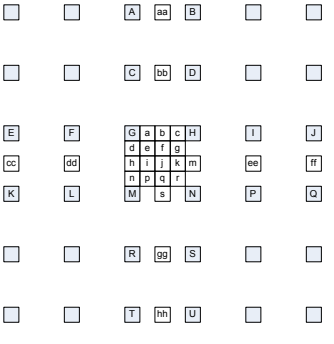


Fig. 1. Illustration of half-pixel samples. Upper-case letters indicate samples on the full-sample grid, while lower letters indicate samples in between at fractional-sample positions.

$$b = \frac{(G + H + 1)}{2}, \quad h = \frac{(G + M + 1)}{2} \qquad (1)$$
$$j = \frac{(G + H + M + N + 2)}{4}$$

$$b_1 = (E - 5F + 20G + 20H - 5I + J) \qquad (2)$$
$$h_1 = (A - 5C + 20G + 20M - 5R + T)$$

$$b = (b_1 + 16) \gg 5 \qquad (3)$$
$$h = (h_1 + 16) \gg 5$$

$$j_1 = cc - 5dd + 20h_1 + 20m_1 - 5ee + ff \qquad (4)$$
$$j = (j_1 + 512) \gg 10$$

In H.264, half-pixel values are obtained by applying a one-dimensional 6-tap filter horizontally and vertically [1][6]. In Fig. 1, H-halfpixel, b, and V-halfpixel, h, are derived by first calculating the intermediate values b_1 and h_1, respectively, then applying (2)'s 6-tap filter and obtaining the final values by (3). In the case of, HV-halfpixel, j is obtained by (4) where intermediate values denoted as cc, dd, ee, m_1 and ff are obtained in a manner similar to h_1.In [7], the percentage of half-pixel MVs is 37.03%. MVs with half-pixel are a significant part of all MVs, therefore, the DCT-domain transcoder must have a half-pixel correction filter.

3 Half-Pixel Correction in DCT-Domain

Fig. 2 shows a block diagram of the proposed transcoder architecture for MPEG-2 to H.264/AVC where Q_1 and Q_2 denote MPEG-2 and H.264 quantization and q_1 and q_2 denote Qstep, respectively. This paper focuses on half-pixel correction, therefore, we give the full details of CHPD and DCT_Conv.

Lowercase variables indicate pixel based values, whereas the corresponding uppercase variables are the DCT representation of the lowercase variables.

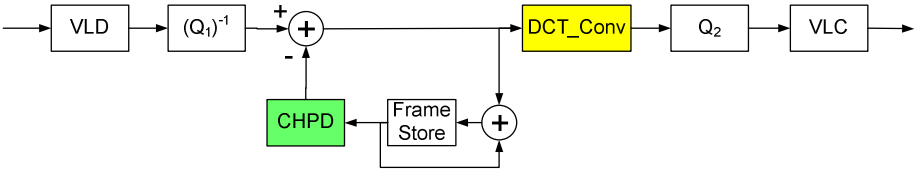


Fig. 2. Illustration of the proposed transcoder architecture for MPEG-2 to H.264/AVC

3.1 Proposed Schemes

The proposed CHPD consists of 3 phases: the first is IMC (Inverse Motion Compensation), the second is calculating the difference between the half-pixel values from the standards, and the third is adding the correction block to the input block. To obtain a reference frame, IMC uses the DCT reference frame in Frame Store [7][8]. The

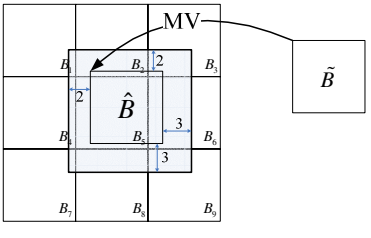


Fig. 3. The proposed IMC (Input block is overlapped 9 blocks)

conventional IMC extracts an 8×8 block to obtain MPEG-2 half-pixel values [7], but the proposed IMC extracts a maximum 13×13 block instead because the H.264 filter needs more pixels, for example left 2 pixels, right 3 pixels, upper 2 pixels, and bottom 3 pixels. As we can see in Fig. 1, the half-pixels located on the border of the 8×8 block need more than two or three pixels to calculate HV-halfpixel. Fig. 3 shows that a 13×13 block used to obtain the correction block is overlap by a maximum of 9 blocks.

The calculation, $\hat{b}$, is formalized as

$$\begin{aligned}\hat{b} &= \sum_{i=1}^{n} e_l \cdot b_i \cdot e_r \quad , \; where \;\; n \in S, \quad S = \{2,3,4,6,9\} \\ &= \underbrace{\begin{pmatrix} b_1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix}}_{\hat{b}_1} + \underbrace{\begin{pmatrix} 0 & b_2 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix}}_{\hat{b}_2} + \cdots + \underbrace{\begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & b_n \end{pmatrix}}_{\hat{b}_n}\end{aligned} \tag{5}$$

where n is the number of overlapped blocks and b_i is an 8×8 reference block but the size of e_l , e_r and $\hat{b}$ change according to the half-pixel direction, e_l and e_r are shift matrices and are defined below.

$$e_l = \begin{pmatrix} 0 & I_{l\times l} \\ 0 & 0 \end{pmatrix}_{m'\times n'} \quad or \quad \begin{pmatrix} 0 & 0 \\ I_{l\times l} & 0 \end{pmatrix}_{m'\times n'}, \quad e_r = \begin{pmatrix} I_{l\times l} & 0 \\ 0 & 0 \end{pmatrix}_{m''\times n''} \quad or \quad \begin{pmatrix} 0 & 0 \\ 0 & I_{l\times l} \end{pmatrix}_{m''\times n''} \tag{6}$$

$$, where \quad m', n', m'', n'' \in \{8,13\}$$

For example, the size of $\hat{b}$, e_l and e_r , V-halfpixels are 13×8, 13×8, and 8×8,respectively

Due to the distributive property of the DCT, we can get (7) from (5).

$$\hat{B} = \sum_{i=1}^{n} E_L \cdot B_i \cdot E_R \quad , where \; n \in S, \quad S = \{2,3,4,6,9\} \tag{7}$$

Let us find the difference of the half-pixels between the two standards by (8),

$$\begin{cases} C = F_L^S \cdot \hat{B} \gg 5 \\ C = \hat{B} \cdot F_R^S \gg 5 \\ C = F_{HL} \cdot \hat{B} \cdot F_{HR} \gg 10 \; - F_{ML} \cdot \hat{B} \cdot F_{MR} \gg 2 \end{cases} \tag{8}$$

$$f_{hl} = \begin{pmatrix} 1 & -5 & 20 & 20 & -5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -5 & 20 & 20 & -5 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -5 & 20 & 20 & -5 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -5 & 20 & 20 & -5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -5 & 20 & 20 & -5 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -5 & 20 & 20 & -5 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -5 & 20 & 20 & -5 & 1 \end{pmatrix} \quad f_{ml} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

where f is a matrix of constant values, $F_L^S = F_{HL} - 16 \cdot F_{ML}$, $F_R^S = F_{HR} - 16 \cdot F_{MR}$, $f_{hr} = f_{hl}^T$ and $f_{mr} = f_{ml}^T$.

Equation (8) is divided into three parts such as, V-halfpixel, H-halfpixel and HV-halfpixel according to half-pixel direction,
To reduce complexity, we use (9) instead of (7), and (8) for V-halfpixel and H-halfpixel because E_L, E_R, F_L^S, F_R^S are matrices of constant values

$$\begin{aligned} C &= \sum_{i=1}^{n} W_L \cdot B_i \quad , where \quad n \in S, \quad S = \{2,3,4,6,9\} \\ C &= \sum_{i=1}^{n} B_i \cdot W_R \quad , where \quad n \in S, \quad S = \{2,3,4,6,9\} \end{aligned} \tag{9}$$

$$W_L = F_L \cdot E_L \gg 5, \qquad W_R = E_R \cdot F_R \gg 5$$

Then correction block (C) is added to input block ($\tilde{B}$) as follows:

$$N = \tilde{B} - C \tag{10}$$

At last, we derive a new residual block from (10) in the DCT-domain and then the new residual block (N) is used as an input into DCT_Conv to adjust the transform between the two standards.

3.2 DCT Conversion

H.264 adopts a 4×4 integer transform instead of the 8×8 float transform used in MPEG-2, H.263 and MPEG-4, so the transcoder must convert an 8×8 float DCT block into four 4×4 integer DCT blocks as shown in Fig. 4. DCT Conversion is performed once by the matrix operation defined as:

$$\begin{aligned} \hat{N}_i &= h_I \cdot n_i \cdot h_I^T \qquad (1 \le i \le 4) \\ &= h_I \cdot (l_i \cdot n \cdot r_i) \cdot h_I^T \\ &= (h_I \cdot l_i) \cdot n \cdot (r_i \cdot h_I^T) \\ &= (h_I \cdot l_i) \cdot (h_F^T \cdot N \cdot h_F) \cdot (r_i \cdot h_I^T) \\ &= (h_I \cdot l_i \cdot h_F^T) \cdot N \cdot (h_F \cdot r_i \cdot h_I^T) \\ &= u_I \cdot N \cdot u_r \end{aligned} \tag{11}$$

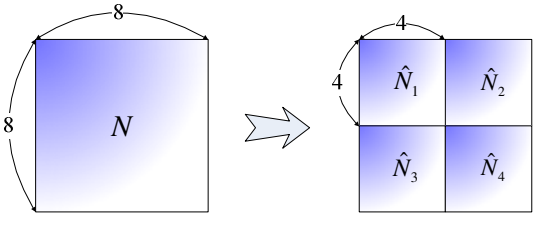


Fig. 4. Extracting 4×4 sub-blocks from an 8×8 block in the transcoder. N and $\hat{N}_i$ denote an 8×8 float DCT block, and 4×4 integer DCT sub-block, respectively

where $h_I \cdot l_i \cdot h_F^T (= u_i)$, $h_F \cdot r_i \cdot h_I^T (= u_r)$, h_I and h_F denote the 4×4 integer transform matrix, and 8×8 float transform matrix, respectively. 4×8 matrix l_i and 8×4 r_i are defined in (12), where $I_{4\times4}$ is an identity matrix of size 4. Refer to [9] for a more detailed explanations.

$$l_0 = l_1 = [I_{4\times4} \quad 0]_{4\times8}, \quad l_2 = l_3 = [0 \quad I_{4\times4}]_{4\times8}$$
$$r_0 = r_2 = \begin{pmatrix} I_{4\times4} \\ 0 \end{pmatrix}_{8\times4}, \quad r_1 = r_3 = \begin{pmatrix} 0 \\ I_{4\times4} \end{pmatrix}_{8\times4} \tag{12}$$

4 Experimental Results

In this section, we compare the performance of our proposed scheme with that of the PDT as shown in Fig. 5, because DDT that performs half-pixel correction does not exist currently. To precisely simulate different transcoding processes, we have implemented the architecture shown in Fig. 2 by modifying the MPEG-2 TM5 [10] and by partially adopting the H.264 JM8 [11]. Test video sequences are constrained by a GOP structure of (N = 15, M = 1) and at a constant frame rate of 30.

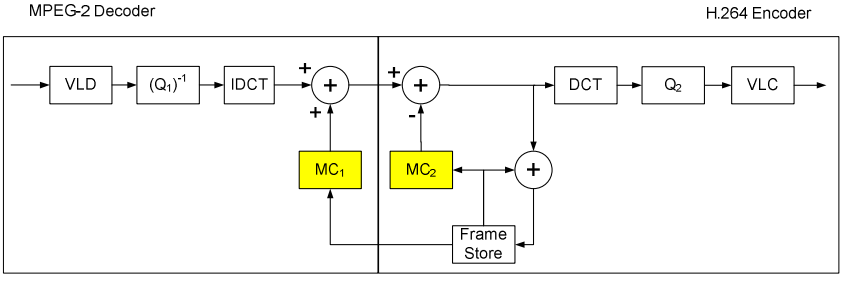


Fig. 5. The PDT (Open-loop Pixel-Domain Transcoder) architecture

Fig. 6 shows the average PSNRs of the frames ranging from 1 to 100 for the Football, and the Foreman sequences. In these figures, we can see that, as a whole, the proposed scheme performs better than that of the PDT. Table 1 denotes the average PSNR of the first 100 frames for Football and Foreman, and it shows that the proposed method outperforms PDT by 0.22 dB. As we can see in Fig. 5, PDT has two MCs, MC_1 and MC_2, which calculate MPEG-2, and H.264 half-pixel values, which lead to rounding errors. The proposed method reduces rounding errors because it directly calculates differences using matrix operations, and it avoids the DCT and IDCT mismatch problem of PDT. These are the reasons the proposed method outperforms PDT. Now, let's examine the real images as reconstructed by both transcoders. Fig. 7 compares in subjective quality. For example, the right person's leg of (a) shows better quality than that of (b).

Table 1. Illustration of the average PSNRs with 100 frames

	PDT		Proposed(CHPD)	
	$q_1 = 5\ q_2 = 10$	$q_1 = 5\ q_2 = 15$	$q_1 = 5\ q_2 = 10$	$q_1 = 5\ q_2 = 15$
Football	29.4	28.3	29.6	28.6
Foreman	31.1	29.5	31.3	29.7

Table 2 shows transcoding times for both methods without VLD, Q_1, Q_2 and VLC. The table shows that sequences with little activity, such as Claire, News, and, Akiyo, take a shorter time to transcoder than they did in PDT, whereas the sequences with high activity, such as Foreman takes a longer time than that of PDT due to high computational matrix operations. Accordingly, the proposed method is more appropriate for transcoding video already compressed by MPEG-2 to high quality video, H.264, offline, however, it will be possible to transcode in real time in the near future.

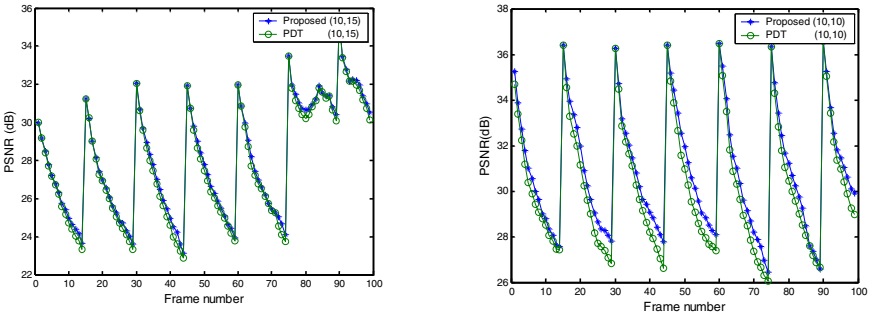


Fig. 6. The PSNR performance for the sequence from frame 1 to 100. (a) For Football where q_1 and q_2 are set at 10 and 15, respectively. (b) For Foreman where q_1 and q_2 are set at 10.

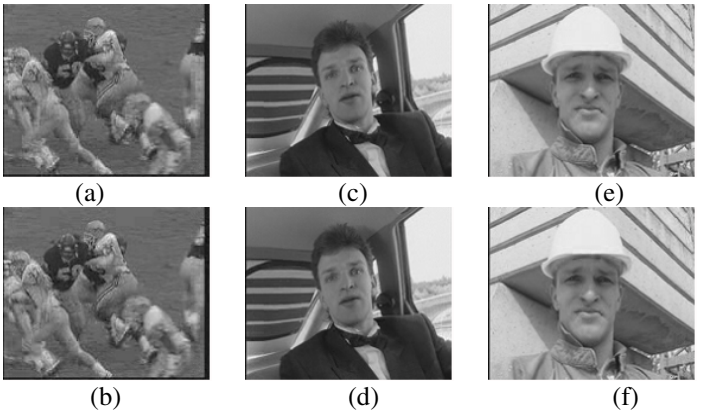


Fig. 7. The real images for the sample video sequences at q_1 and q_2 = 10: Football (a)(b); Carphone (c)(d); The PDT images are arranged in the first row, and the proposed method images are in the second row.

Table 2. Transcoding time of both methods (ms)

	Claire	News	Akiyo	Foreman
PDT	32.91	33.80	25.13	148.73
Proposed	30.15	32.45	24.81	206.94

5 Conclusions

In this paper, we proposed a half-pixel correction method for DCT-domain MPEG-2 to H.264 transcoding. The proposed method consists of CHPD and DCT_Conv. The former adds the input block to a correction block that contains the differences between two standards. The latter converts the 8×8 float DCT block into four 4×4 integer DCT blocks. The matrices are stored in memory to reduce computational complexity because they contain fixed constant values. The proposed method improves the video quality by reducing the MC phase. As far as we know, our method is the first correction scheme for the DCT-domain H.264 transcoding architecture. Our next research is to reduce the computational complexity for high activity video sequences.

Reference

1. T.Wiegand, G.J.Sullivan, G. Bjontegaard, and A.Luthra, "Overview of the H.264/AVC video coding standard,"IEEE Trans. Circuits Syst. Video Technol.,vol. 13, pp. 560-576, July 2003
2. J. Ostermann, J.Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video Coding with H.264/AVC: Tools, Performance, and Complexity", IEEE Circuits Syst. Magazine, vol.4, no.1, pp.7-28, Apr. 2004.
3. Hari Kalva. "Issues in H.264/MPEG-2 Video Transcoding". Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE, 5-8 Jan. 2004
4. A. Vetro, C. Christopoulos, and H. Sun, "Video Transcoding Architectures and Techniques: An Overview," IEEE Signal Processing Magazin, vol. 20,no.2,p.18-29, March2003
5. ISO/IEC 13818-2:1995(E) pp.83~100
6. Iain E.G. Richardson, "H.264 and MPEG-4 Video Compression" WILEY , 2003
7. G.Cao, Z.Lei, J.Li, N.D.Georganas, Z.Zhu. "A Novel DCT Domain Transcoder for Transcoding Video Streams with Half-pixel Motion Vectors" Real-Time Imaging (Elsevier Science) Vol.10 Issue 5, Oc 2004, pp 331-337
8. T. Shanableh and M. Ghanbari,"Hybrid DCT/pixel domain architecture for heterogeneous video transcoding" Signal processing: Image Communication, vol.18, pp.601-620, 2003
9. Quantization/DCT Conversion Scheme for DCT-domain MPEG-2 to H.264/AVC Transcoding, IEICE Trans. Communication, July, 2005 accepted
10. http://diml.yonsei.ac.kr/~wizard97/mpeg2/mpeg2v12.zip
11. http://bs.hhi.de/~suehring/tml/download/old_jm/jm82.zip

An Object Interface for Interoperability of Image Processing Parallel Library in a Distributed Environment

Andrea Clematis, Daniele D'Agostino, and Antonella Galizia

IMATI-CNR, Via de Marini 6, 16149 Genova, Italy
{antonella, clematis, dago}@ge.imati.cnr.it

Abstract. Image processing applications are computing demanding and since a long time much attention has been paid to the use of parallel processing. Emerging distributed and Grid based architectures represent new and well suited platforms that promise the availability of the required computational power. In this direction image processing has to evolve to heterogeneous environments, and a crucial aspect is represented by the interoperability and reuse of available and high performance code. This paper describes our experience in the development of PIMA(GE)^2, Parallel IMAGE processing GEnoa server, obtained wrapping a library using the CORBA framework. Our aim is to obtain a high level of flexibility and dynamicity in the server architecture with a possible limited overhead. The design of a hierarchy of image processing operation objects and the development of the server interface are discussed.

1 Introduction

During the last decades Image Processing has become a topic of interest for a broad scientific community, since the emerging digital technologies allow to process complex signals; on the other hand image processing growth is empowered by its applications, covering several fields of interest such as medicine, industries, military, meteorology, and earth observation. The variety of problems arising with the different application domains leads to the evolution of several specific analysis and processing techniques, that mainly differ in the type of images they are related with, or in the task they should perform. However capturing, processing, and managing images require an enormous computational effort, often too high for a single processor architecture. The demand for computing resources is increasing and distributed computing environments provide an effective and attractive solution. The emerging technologies to overcome these requirements lead to distributed architectures and Computational Grids [8].

To satisfy image processing requirements, it is important to explore the use of Computational Grids to allow the resolution of computing intensive applications and at the same time preserving previously developed software. In this direction a great interest is dedicated to interoperability and reuse of legacy code. In the recent past different technologies and methodologies have been developed to enable this process. The generally adopted solution is the legacy code

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 584–591, 2005.

encapsulation: the code is left in its environment and dynamically connected to the new technologies through a wrapper, allowing the software performs in a Client/Server system [20]. In this way it is possible to profit of the advantages offered by the new infrastructures and to keep previously developed softwares, still representing a useful support to solve problems.

In the presented experience we encapsulate a parallel image processing library in a component framework, CORBA [3], in order to develop PIMA(GE)^2, Parallel IMAGE processing GEnoa distributed and heterogeneous server. The applications developed using PIMA(GE)^2 may be integrated in a Grid architecture. In particular the server implements the most common image processing operations performed in a data parallel fashion; the parallelism is hidden from the users and totally managed by the server, that also applies a transparent optimization policy. The legacy code, i.e. the library, is totally reused.
The server also represents a way to combine Single Program Multiple Data, SPMD, parallel computing model implemented with MPI [13], and distributed programming technologies, using CORBA, without modifications to the Object Management Group [15], OMG standard.

In the paper the main emphasis is given to the definition of the PIMA(GE)^2 interface, that represents in our opinion, one of the most important elements in this kind of work, in order to simplify the reuse, ensuring flexibility and interoperability. A particular interesting aspect is that legacy code is a parallel image processing library. This is kept in due consideration during the interface planning and the architecture development.

The paper is organized as follow: in the next Sect. the evolution of the library concept in a distributed and heterogeneous environment is discussed; in Sect. 3 the wrapping phase and the definition of the Application Programming Interface (API) of the server are described; in Sect. 4 the PIMA(GE)^2 architecture is briefly detailed; we present experimental results and conclusions in Sect. 5.

2 Moving Towards Heterogeneous Dynamic Image Processing

In order to allow the use of image processing tools in distributed and Grid environments, it is necessary to evolve from library to a component based server. We have to keep the good features of the library and add to them other important characteristics, such as interoperability, and performance portability, to permit the integration in distributed and heterogeneous environments. Our aim is to obtain a high level of flexibility and dynamicity in the server architecture with an acceptable overhead.

The parallel image processing library, used during this work, is a still ongoing project implemented in C++ and MPI [9]. The library implements the most common image processing operations, according to the classification provided in Image Algebra [18]. The operations are performed in a data parallel fashion; the parallelism is hidden from the users and totally managed by the library. It also provides a performance optimization oriented to different levels: an optimization

policy is applied to perform a suitable management of communication and memory operations; the data distribution is oriented to obtain load balancing; and others ongoing efforts are dedicated to the exploitation of memory hierarchies. Also the optimization aspects are transparent for the users.

A library is often considered as a static tool, designed to develop the user's own applications; it is mainly designed to be executed on an homogeneous architecture in order to develop static applications. However in the era of distributed and Grid computing, the concept of library is evolving. Moving the software towards distributed, dynamic and heterogeneous environments, we are transforming it in a component-based bundle that can be accessed using a server based interaction. It means that the functions performed by the library are evolved into components, and an application becomes a dynamic components concatenation, executed on an heterogeneous infrastructures. The server is no longer a static entity, it is able to manage multiple requests performed by different distributed clients.

Since the library is not modified and totally reused during the encapsulation process, the server inherits its features, such as transparence, high performance and completeness. These features should be improved by adding further properties, such as distributed and heterogeneous executions, software extensibility and interoperability, dynamicity and portability on the Grid. We obtain a dynamic interaction model that permits to accommodate request arriving from heterogeneous Client in a distributed environment.

2.1 Related Works

In the image processing domain it is possible to find different examples of parallel libraries, for example VSIPL++ [12], ParHorus [19], PIPT [17]. They provide object-oriented image processing code, and ensure high performance executions; in fact they are improved with performance optimization policy and are compliant to the standard features ensured by a scientific library. There are also several on-going efforts aiming at the exploitation of distributed and heterogeneous environments reusing the previous developed software. Important and successful projects are represented by VGrADS [10], Cactus [2], NetSolve [14], that aim to the integration of numerical libraries in Grid environments. The need of migration to distributed and Grid environments is even present in the image processing community, as different and actual works show, [4, 1, 11].

3 The Definition of the PIMA(GE)^2 Interface

To allow a simple and coherent use of the library in a distributed environment, the most relevant element is the definition of effective and flexible interfaces, that permit the development of efficient image processing applications. It can be achieved through a natural evolution from the library legacy code to the PIMA(GE)^2 server; this step will directly provide a model to derive interfaces with the mentioned properties. The crucial element to achieve this goal becomes

a classification and a hierarchical organization of the operations implemented by the library. Our idea is to group them in different objects and then outline a hierarchy of image processing operation objects. This effort is useful to provide the model for the definition of an adequate PIMA(GE)^2 interface, in fact this structure will be codified using the CORBA Interface Description Language (IDL).

3.1 The Legacy Code Classification

During the classification of the library operations, we introduce conceptual objects in order to group together functions according to different rules. We consider one of distinguishing marks the similar nature or behavior of the operations; for example we grouped in the object *Unary Operation* the functions applied to individual pixels, i.e. square root, absolute value, etc. Another rule is the belonging to the same specialization field, for example in the object *Differential Operations* are grouped the differential operators, i.e. Gradient, Laplacian, Hessian; or we group together functions with the same data structures in input, for example functions changing the image geometry, i.e. rotation, translation and scaling, are grouped in the object *Geometric Operations*.
The result of the classification step is a set of image processing conceptual objects, made up of eight elements. They are not intended to prescribe how an operation is performed but to underline the operation similarity and to help in the definition of an effective and efficient interface. The classification does not put constraints on the code implementation and does not imply any code modification, because it is a conceptual step aimed to obtain an object oriented structure of the library code.

3.2 The Hierarchical Library Organization

Figure 1 represents the hierarchial library structure. The bottom layer of the hierarchy is represented by the main data structures of the library, that is the structures used to store images, convolution kernels, geometric matrices. We relate to them functions for memory management, I/O operations, and generic operations when it is possible, for example to transpose or invert a MATRIX. In this way we obtain three objects representing the basic elements of the server; they already include, as methods, some of the objects derived by the library classification step.

The second level of the PIMA(GE)^2 server is a set of five objects, obtained by grouping together the library objects, according to the rules already mentioned. In this level we have **Point Operations** that take in input only one IMAGE object; **Image Arithmetic** Operations that take in input two IMAGE objects; **Geometric Operations** that take in input one IMAGE object and one MATRIX object; **Convolution Operations** that take in input one IMAGE object and one KERNEL object; **Differential Operations** that take in input one IMAGE and perform differential operators.

The top layer of the server is represented by the object **Operations**, it contains the image processing operations performed by the server and groups

together the objects already described. Figure 1 explicitly visualizes the developed hierarchy and the inter-dependences among objects. The objects allow an easier management of the library operations, since a client is not more involved with a large number of functions, but he has to consider and to handle a small set of clear and well-defined objects.

3.3 The PIMA(GE)^2 Server Interface

Since the hierarchy definition is completed, the PIMA(GE)^2 interface implementing phase is totally planned and easily obtained. In fact the aim of the object hierarchy is to drive the definition of an adequate CORBA interface, and its IDL based implementation. Such interface represents the PIMA(GE)^2 API that will be called by Client applications. In this way the server provides a sequential API that hides the different levels and the computational complexity: underlying heterogeneous architectures, optimization policies, and parallel programming level. From the client point of view, the interface looks like a standard CORBA server, in fact the parallelism is totally hidden to the client, as well as the mechanisms and policies applied to allow MPI-CORBA compatibility, described in the next Sect.

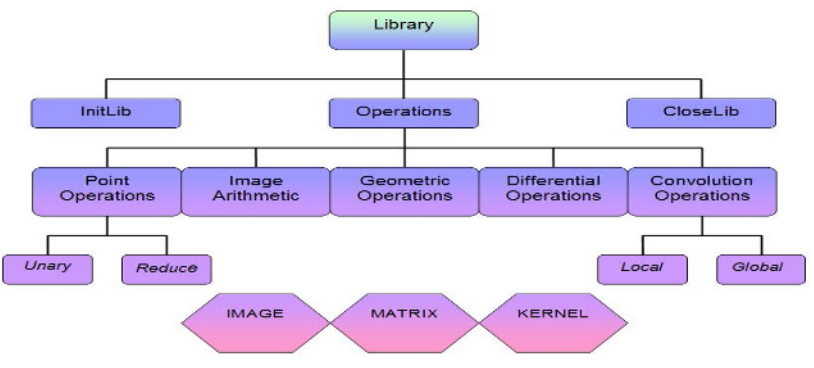


Fig. 1. Overview of the PIMA(GE)^2 hierarchy

4 The Server Architecture

The main difficulty in the PIMA(GE)^2 implementation rises from the presence of parallelism in the computation. It imposes the management of two different environments: the CORBA framework (server side) and the MPI library (legacy code side). They are aimed to manage different kind of problems, therefore there is not a standard schema to allow their cooperation. In fact CORBA was born to develop mainly sequential applications, hence it does not support intrinsic compatibility with any kind of parallel environment. The use of CORBA to wrap parallel applications has been considered in different works, through the

definition of parallel CORBA objects, [6, 16], or component environments built on CORBA [7, 5]. They all require the modification of CORBA middleware and hence are not standard solutions.

We decided to use a standard CORBA implementation, TAO [21], and look for a way allowing the development of parallel computations inside CORBA object. In order to achieve this goal we designed a special process to perform specific tasks. This process acts as the gateway that coordinates the two environments, allowing the computation. It has a dual position in the software architecture , in fact it will be in the same time a CORBA and a MPI process. On the CORBA side, it has to activate the ORB and perform as a CORBA server; at the same time it is also one of the spawned MPI processes. Acting as a MPI process, it will manage the parallel computation. This solution is possible because, acting as server CORBA, the designated process knows the client requirements an d can communicate them to the others MPI processes, can coordinate the computation and give back the informations to the clients. This solution does not require modifications to the legacy code, since it exploits the presence, in the legacy code, of the coordinator MPI process during the initialization and I/O phases. The behavior is explained in the Fig. 2.

5 Experimenting the Approach and Conclusions

The experimental results we obtained show a reasonable overhead due to the presence of the CORBA wrapping framework. We considered an edge detection algorithm that calls different parallel library functions, and carried out three tests:

1. The parallel library is used in a normal MPI environment (i.e. without PIMA(GE)^2);
2. PIMA(GE)^2 is extended with the new algorithm and a remote client calls the new function;
3. A remote client calls the sequence of functions of PIMA(GE)^2 that implements the algorithm.

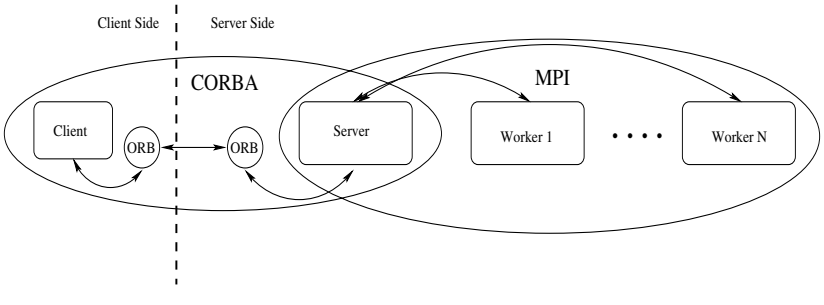


Fig. 2. The CORBA-MPI interaction

Table 1. Experimental results, times are given in seconds

Processes	1	2	3	4	6	8
C++ MPI library	1037.624	564.744	407.585	326.135	243.534	208.834
PIMA(GE)^2 one call	1038.668	566.866	409.499	328.496	244.700	209.923
PIMA(GE)^2 sequence	1039.800	568.914	411.098	332.489	248.363	212.012

Some general considerations apply to the three tests. In all the cases the parallel execution environment is a Linux cluster with eight nodes interconnected by a Gigabit switched Ethernet, and each node is equipped with a 2.66 GHz Pentium processor, 512 Mbytes of RAM and two EIDE disks interface in RAID 0. The remote client is a PC that does not belong to the cluster but is connected on the same LAN with a 10/100 Ethernet link. We avoid to use a "true" remote client, since we would like to measure the overhead due to PIMA(GE)^2 infrastructure, without the possible unpredictable behaviour of an actual remote interconnection.

The same optimisation policy is used within the library in the three cases, in order to minimize the set-up operations, to optimise data distribution and to reduce data movement. Finally no actual images, but only symbolic identifiers are transmitted along the ORB. This decision is motivated by the fact that large size data transfer between the client and the server can be more efficiently accomplished in other way, like using optimised FTP protocols.

The three tests permit us to assess the performance quality of PIMA(GE)^2. Table 1 reports execution times in second for the edge detection algorithm on 1 MByte image in the three cases and for different number of processors. The first test is necessary in order to assess the performance of the library and to define a term of comparison for PIMA(GE)^2. With the second test we measured the overhead due to the CORBA set-up. As expected this set-up is a fixed time (about 2-3 seconds in our experiments) that does not depend on the number of parallel processes. In this case the interaction between the client and the server is very coarse grained and we have a minimum number of communications on the ORB. The third test is aimed to measure the performance degradation due to a more consistent use of the ORB communication channel. In fact in front of a single call of experiment 2, we have in this case a number of about 2166 calls and each one corresponds to a double communications on the ORB. Also in this case the overhead is quite limited especially for compute intensive applications. In this paper we have reported our experience in the design of PIMA(GE)^2 server interface. Future developments will be aimed to the integration of this CORBA based implementation with other Grid middleware.

Acknowledgments

This work has been supported by MIUR program L. 449/97-00 "High Performance Distributed Platform", by FIRB strategic project Grid.it, and by CNR strategic project @SWING.

References

[1] Barros Jr, E.M., Shen, M., On Wangenheim, A.: A model for distributed medical image processing using CORBA. IEEE CNBS (2001) 189-

[2] The Cactus Project home page, www.cactuscode.org/

[3] The CORBA home page, http://www.corba.org/

[4] De Alfonso, C., Blanquer, I., Herniández, V.: Providing with High Performance 3D Medical Image Processing on a Distributed environment. Health GRID (2003)

[5] Dennis, A., Peréz C., Priol, T.: PadicoTM: An open integration framework for communication middleware and runtimes. IEEE Intl. CCGrid (2002) 144-151

[6] Dennis, A., Peréz C., Priol, T.: Achieving Portable and Efficient Parallel CORBA Objects. Concurrency and Computation: Practice and Experience, **15** 10 (2003) 891-909

[7] Dennis, A., Peréz C., Ribes, A.: Padico: a component-based software infrastructure for Grid Computing. IPDPS (2003) 2

[8] Forster, I., Kesselman, C.: The grid: blueprint for a new computing infrastructure. 2nd Edition Morgan Kaufmann (2004)

[9] Galizia, A.: Evaluation of optimization policies in the impletation of Parallel Libraries. Technical Report IMATI-CNR-Ge **20** (2004)

[10] GrADS Project Home Page, http://www.hipersoft.rice.edu/grads/

[11] Hastings, S., Kurc, T., Langella, S., Catalyurek, U., Pan T., Saltz, J.: Image Processing for the Grid: a toolkit for building Grid-enable Image Processing Applications. IEEE/AMc CCGRID (2003) 36-43

[12] Lebak, J., Kepner, J., Hoffmann, H., Rudtledge, E.: Parallel VSIPL++: an open standard library for high-performance parallel signal processing. IEEE Proceedings **93** 2 February (2005) 313-330

[13] MPI Home Page, http://www-unix.mcs.anl.gov/mpi/

[14] NetSolve Home Page, http://icl.cs.utk.edu/netsolve/

[15] OMG Official Website, http://omg.org

[16] Peréz, C., Priol, T., Ribes, A.: A Parallel CORBA Component Model. INRIA Technical Report N.4552 September (2002)

[17] PIPT Home Page, http://www.osl.iu.edu/research/pipt

[18] Ritter, G., Wilson, J.: Handbook of Computer Vision Algorithms in Image Algebra. 2nd edition CRC Press Inc (2001)

[19] Seinstra, F., Koelma, D., Geusebroek, J.M.: A software architecture for user transparent parallel image processing. Parallel Computing **28** 7-8 (2002) 967-993

[20] Sneed, H.M.: Encapsulation of Legacy Software: A technique for reuse software components. Annals of Software Engineering **9** (2000) 293-313

[21] Tao home page, http://www.cs.wustl.edu/~schmidt/TAO.html

Markovian Energy-Based Computer Vision Algorithms on Graphics Hardware

Pierre-Marc Jodoin, Max Mignotte, and Jean-François St-Amour

Université de Montréal, DIRO,
P.O. Box 6128, Studio Centre-Ville, Montréal, Québec, H3C 3J7
{jodoinp, mignotte, stamourj}@iro.umontreal.ca

Abstract. This paper shows how Markovian segmentation algorithms used to solve well known computer vision problems such as *motion estimation*, *motion detection* and *stereovision* can be significantly accelerated when implemented on programmable graphics hardware. More precisely, this contribution exposes how the parallel abilities of a standard Graphics Processing Unit (usually devoted to image synthesis) can be used to infer the labels of a label field. The computer vision problems addressed in this paper are solved in the maximum a posteriori (MAP) sense with an optimization algorithm such as ICM or simulated annealing. To do so, the *fragment processor* is used to update in parallel every labels of the segmentation map while rendering passes and graphics textures are used to simulate optimization iterations. Results show that impressive acceleration factors can be reached, especially when the size of the scene, the number of labels or the number of iterations is large. Hardware results have been obtained with programs running on a mid-end affordable graphics card.

1 Introduction

Graphics hardware nowadays available are often equipped with a so called *Graphics Processing Unit* (GPU). This unit can execute general purpose programs independently of the CPU and the central memory. As the name implies, this architecture was optimized to solve typical graphics problems in the goal of rendering complex scenes in real-time. Because of the very nature of conventional graphics scenes, graphics hardware have been designed to efficiently manipulate texture, vertices and pixels. These primitives are processed either by the *vertex processor* or the *fragment processor*. What makes these processors so efficient is their fundamental ability to process vertices and fragments (see pixels) in parallel, involving interesting acceleration factors.

However, in spite of appearances, it is possible to take advantage of the parallel abilities of programmable graphics hardware to solve problems that goes beyond graphics. This is what people call *general-purpose computation on GPUs* (GPGPU). Some authors have shown that applications such as fast Fourier transforms [1], linear algebra [2], motion estimation and spatial segmentation could run on graphics hardware [3,4]. Even if these applications have little in common

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 592–603, 2005.

with traditional computer graphics, they all share a common denominator: they are problems solved by parallizable algorithms.

This paper presents how Markovian segmentation algorithms used to solve computer vision problems such as motion detection [5], motion estimation [6] and stereovision [7], can be significantly accelerated when implemented on a typical GPU. These computer vision problems are herein expressed by Markovian energy-based models through Gibbs distribution. This framework stipulates that a solution (also called *label field* or *segmentation map*) is *optimal* when it minimizes a global energy function made of a *likelihood* term and a *prior* term[8].

For all Markovian energy-based model, the label field has to be inferred by an optimization algorithm such as simulated annealing (SA) [9], ICM [10] or HFC [11]. Although ICM and HFC are much faster than SA, the processing time of these deterministic optimization algorithms can be very much prohibitive. In this contribution, we expose how optimizers such as SA or ICM –used to solve energy-based computer vision problems– can be significantly accelerated when implemented on programmable graphics hardware. Even if GPUs are cutting edge technologies made for graphics rendering, implementing a MAP segmentation algorithm on a fragment processor isn't much more difficult than writing it in a typical C-like procedural language.

The remainder of the paper is organized as follows. In Section 2, a review of the Markovian theory is proposed before Section 3 presents the three computer vision problems we are interested into. Section 4 presents the optimization algorithms ICM and SA after which Section 5 gives a look to the graphics hardware architecture. Finally, Section 6 shows some experimental results before Section 7 concludes.

2 Markovian Segmentation

The computer vision problems this contribution tackles can be seen as segmentation problems. As a matter of fact, these vision problems aim at subdividing observed input images into uniform regions by grouping pixels having high-level features in common such as motion or depth. Starting with some observed data Y (which is typically one or more input images), the goal of any segmentation process is to infer a label field X containing the class labels (i.e. labels indicating whether a pixel belongs or not to a moving area or a certain depth). In computer vision, X and Y are generally defined over a rectangular lattice of size $\mathcal{N} \times \mathcal{M}$ represented by $S = \{s | 0 \leq s < \mathcal{N} \times \mathcal{M}\}$ where s is a site located at the Cartesian position (i, j) (for simplicity, s is sometimes defined as a pixel). It is common to represent by a low-case variable such as x or y, a realization of the label field or the observation field. For each site $s \in S$, its corresponding element x_s in the label field takes a value in $\Gamma = \{e_1, e_2, ..., e_N\}$ where N is the total number of classes. In the case of motion detection for example, N can be set to 2 and $\Gamma = \{\text{StaticClass,MobileClass}\}$. Similarly, the observed value y_s takes a value in $\Lambda = \{\epsilon_1, \epsilon_2, ..., \epsilon_\zeta\}$ where ζ is set to 2^8 for gray-scale images and 2^{24} for color

images. In short, a segmentation model is made of an observation field y that is to be decomposed into N classes by inferring a label field x.

In the context of this paper, the goal is to find an *optimal* labeling $\hat{x}$ which maximizes the a posteriori probability $P(X = x|Y = y)$ (that we represent by $P(x|y)$ for simplicity), also called the *maximum a posteriori* (MAP) estimate [8] : $\hat{x}_{\text{MAP}} = \arg\max_x P(x|y)$. With Bayes theorem, this equation can be rewritten as

$$\hat{x}_{\text{MAP}} = \arg\max_x \frac{P(y|x)P(x)}{P(y)} \tag{1}$$

or equivalently $\hat{x}_{\text{MAP}} = \arg\max_x P(y|x)P(x)$ since $P(y)$ isn't related to x. Assuming that X and Y are Markov Random Fields (MRF) and according to the Hammersley-Clifford theorem [8], the a posteriori probability $P(x|y)$ –as well as the likelihood $P(y|x)$ and the prior $P(x)$– follows a Gibbs distribution, namely

$$P(x|y) = \frac{1}{\lambda_{x|y}} \exp(-U(x, y)) \tag{2}$$

where $\lambda_{x|y}$ is a normalizing constant and $U(x, y)$ is an *energy function*. Combining Eq.(1) and (2), the optimization problem at hand can be formulated as an *energy minimization problem* i.e.:$\hat{x}_{\text{MAP}} = \arg\min_x \big(W(x, y) + V(x)\big)$, where $W(x, y)$ and $V(x)$ are respectively the likelihood and prior energy functions. If we assume that the noise in y isn't correlated, the global energy function $U(x, y)$ can be represented by a sum of local energy functions such as

$$\hat{x}_{\text{MAP}} = \arg\min_x \sum_{s \in S} \left(W_s(x_s, y_s) + V_{\eta_s}(x_s)\right). \tag{3}$$

Here, η_s is the neighborhood around site s and $V_{\eta_s}(x_s) = \sum_{c \in C_s} V_c(x_s)$ is a sum of potential functions defined on so-called cliques c. Function $V_c(x_s)$ defines the relationship between two neighbors defined by c, a binary clique linking a site s to a neighbor r. Notice that $\hat{x}_{\text{MAP}}$ is estimated with an optimization procedure such as SA or ICM which are typically slow algorithms. Details of these algorithms are discussed in Section 4.

3 Computer Vision Problems

3.1 Motion Detection

Among the first paper in computer vision in which a MAP framework was used is the one by Bouthemy and Lalande [5]. In their paper, they proposed a simple energy-based model to solve the problem of motion detection. The solution presented in this Section was inspired of their work.

The goal of motion detection is to segment an animated image sequence into *mobile* and *static* regions. For this kind of application, moving pixels are the ones with a non-zero displacement vector, no matter what direction or speed they might have. Bouthemy and Lalande's [5] paper influenced many subsequent

contributions including the one by Dumontier *et al* [12] who proposed a parallel hardware architecture to detect motion in real time. Unfortunately, the hardware they used was specifically designed and is not, to our knowledge, available on the market.

As for Bouthemy and Lalande [5]'s method, the solution here proposed is based on the concept of temporal gradient and doesn't require the estimation of an optical flow. From two successive frames $f(t)$ and $f(t+1)$, the observation field y is defined as the temporal derivative of the intensity function df/dt namely $y = |f(t+1) - f(t)|$. Assuming that scene illumination variation is small, the likelihood energy function linking the observation field to the label field is defined by the following equation

$$W(x_s, y_s) = \frac{1}{\sigma^2}(y_s - m_p x_s)^2 \tag{4}$$

where m_p is a constant and σ is the variance of the Gaussian noise. Because of the very nature of the problem, $N = 2$ and $x_s \in \{0, 1\}$ where 0 and 1 correspond to static and moving labels. As for the prior energy term, as in [5] and [12], the following Potts model was implemented

$$V_c(x_s) = \begin{cases} 1 & \text{if } x_s \neq x_r \\ -1 & \text{otherwise.} \end{cases} \tag{5}$$

The overall energy function to be minimized is thus defined by

$$U(x, y) = \sum_{s \in S} \Big(\underbrace{\frac{1}{\sigma^2}(y_s - m_p x_s)^2}_{W(x_s, y_s)} + \beta_{\text{MD}} \underbrace{\sum_{c \in \eta_s} V_c(x_s)}_{V_{\eta_s}(x_s)} \Big) \tag{6}$$

where η_s is a second order neighborhood (eight neighbors). Notice that this solution makes the implicit assumption that the camera is still and that moving objects were shot in front of a static background. To help smooth out inter-frame changes, one can add a temporal prior energy term $V_\tau(x_s)$ linking label x_s estimated at time t and the one estimated at time $t-1$.

3.2 Motion Estimation

The goal of motion estimation is to estimate the direction and magnitude of optical motion over each site $s \in S$ of an animated sequence [13,14]. Among the solutions proposed in the literature, many are based on an hypothesis called *lightness consistency*. This hypothesis stipulates that a site $s \in S$ at time t keeps its intensity after it moved at time $t+1$. Although this hypothesis excludes noise, scene illumination variation, and occlusion (and thus is an extreme simplification of the true physical nature of the scene) it allows simple energy functions to generate fairly good results. Under the terms of this hypothesis, the goal of motion estimation is to find, for each site $s \in S$, an optical displacement vector

$\boldsymbol{v}_s$ for which $f_s(t) \approx f_{s+\boldsymbol{v}_s}(t+1)$. In other words, the goal is to find a vector field $\hat{v}$ for which

$$\hat{\boldsymbol{v}}_s = \arg\min_{\boldsymbol{v}_s} |f_s(t) - f_{s+\boldsymbol{v}_s}(t+1)|, \quad \forall s \in S. \tag{7}$$

Notice that the absolute difference could be replaced by a cross-correlation distance. Such strategy is called *region-matching*. In the context of Eq.(7), the observation field y is the input image sequence and $y(t)$ is a frame at time t. The label field x is a vector field made of 2D vectors defined as $x_s = \boldsymbol{v}_s = (\zeta_i, \zeta_j)$ where ζ_i, ζ_j are integers taken between $-d_{max}$ and d_{max} as shown in Fig. 1 (b).

Eq.(7) has one major limitation which comes from the fact that real-world sequences contain textureless areas and/or areas with occlusions. Typically, over these areas more than one vector x_s have a minimum energy, although only one is valid. This is the well known *aperture problem* [15]. In order to guaranty the uniqueness of a consistent solution, several approaches have been proposed [13]. Among these approaches, many add a regularization term (or *smoothness constraints*) whose essential role is to rightly constrain the ill-posed nature of this inverse problem. These constraints typically encourage neighboring vectors to point in the same direction with the same magnitude. In the context of the MAP, these constraints can be expressed as a prior energy function such as the Potts model of Eq.(5). However, since the number of labels can be large (here $(2d_{max}+1)^2$), we empirically observed that a smoother function was better suited. The following linear function was implemented : $V_c(\boldsymbol{x}_s) = \beta_{\text{ME}}\left(|\boldsymbol{x}_s[0] - \boldsymbol{x}_r[0]| + |\boldsymbol{x}_s[1] - \boldsymbol{x}_r[1]|\right)$, where c is a binary clique linking site s to site r. Notice that other smoothing functions are available [15]. The global energy function $U(x,y)$ to be minimized is obtained by combining Eq.(7) to $V_c(\boldsymbol{x}_s)$ as follows

$$U(x,y) = \sum_{s\in S} \Big(\underbrace{|y_s(t) - y_{s+x_s}(t+1)|}_{W_s(x_s,y_s)} + \beta_{\text{ME}} \underbrace{\sum_{\substack{c\in C_s \\ s,r\in c}} \left(|\boldsymbol{x}_s[0] - \boldsymbol{x}_r[0]| + |\boldsymbol{x}_s[1] - \boldsymbol{x}_r[1]|\right)}_{V_{\eta_s(x_s)}} \Big).$$

Let us mentioned that Konrad and Dubois [6] did proposed a similar solution - short of a *line process* they used to help preserve edges.

3.3 Stereovision

The goal of stereovision is to estimate the relative depth of 3D objects from two (or more) images of a scene. For simplicity purposes, many stereovision methods use two images taken by cameras aligned on a linear path with parallel optical axis (this setup is explained in detail in Scharstein and Szelisky's review paper [7]). To simplify the problem, stereovision algorithms often make some assumptions on the true nature of the scene. One common assumption (which is similar to motion estimation's lightness consistency assumption) states that every point visible in one image is also visible (with the same color) in the second

image. Thus, the goal of stereovision algorithms based on such assumption is to estimate the distance between each site s (with coordinate (i, j)) in one image to its corresponding site t (with coordinate $(i + d_s, j)$) in the second image. Such distance is called *disparity* which is, in this case, proportional to the inverse depth of the object projected on site s. This gives rise to a *matching cost* function that measures how good a disparity $d_s \in [0, D_{\text{MAX}}]$ is for a given site s in a reference image y_{ref}. This is expressed mathematically by

$$C(s, d, y) = |y_{\text{ref}}(i, j) - y_{\text{mat}}(i + d_s, j)| \tag{8}$$

where y_{mat} is the second image familiarly called the *matching image*. Notice that the absolute value could be replaced by a quadratic or a robust function [7]. In the context of the MAP, $C(.)$ is the likelihood energy function and the disparity map d is the label field to be estimated. Thus, to ensure uniformity with Section 2's notation, the cost function of Eq.(8) will be defined as $C(s, x, y)$.

To ensure spatial smoothness, two strategies have been traditionally proposed. The first one is to convolute $C(s, x, y)$ with a low-pass filter or a so-called *aggregation filter* w (see [7] for details on aggregation). Although a prefiltering step slows down the segmentation process, it can significantly reduce the effect of noise and thus enhance result quality. Spatial smoothness can also be ensured by adding a prior energy term $V_{\eta_s}(x)$ of the form $V_{\eta_s}(x) = \sum_{s \in S} \sum_{c \in \eta_s} |x_s - x_t|$. Notice that the absolute value could be replaced by another cost function if needed. The global energy function $U(x, y_{\text{ref}}, y_{\text{mat}})$ can thus be written as

$$U(x, y_{\text{ref}}, y_{\text{mat}}) = \sum_{s \in S} \Big(\underbrace{(w * C)(s, x, y)}_{W_s(x_s, y_s)} + \beta_{\text{S}} \underbrace{\sum_{\substack{c \in \eta_s \\ s, r \in c}} |x_s - x_t|)}_{V_{\eta_s}(x_s)} \Big) \tag{9}$$

where β_{S} is a constant. Notice that some authors minimize only the likelihood energy function $W(x, y)$ assuming the low-pass filter is enough to ensure spatial smoothness. This strategy, called *Winner-Take-All* (WTA), is greedy and converges after only one iteration. Another way to save on processing time is to pre-compute the likelihood function in a 3D table. Such table is called the *Disparity Space Integration* (DSI) and contains $\mathcal{N} \times \mathcal{M} \times D_{\text{MAX}}$ cost values.

4 Optimization Procedures

Since Eq.(3) has no analytical solution, $\hat{x}$ has to be estimated with an *optimization* algorithm. An optimization procedure we have implemented is the simulated annealing (SA) which is a stochastic relaxation algorithm similar to the Gibbs sampler. The concept of SA is based on the manner in which some material recrystallize when slowly cooled down after being heated at a high temperature. The final state (called the *frozen* ground state) is reached when temperature gets down to zero. Similarly, SA searches for the global minima by cooling down a temperature factor T [9] from an initial temperature T_{MAX} down to zero. In this

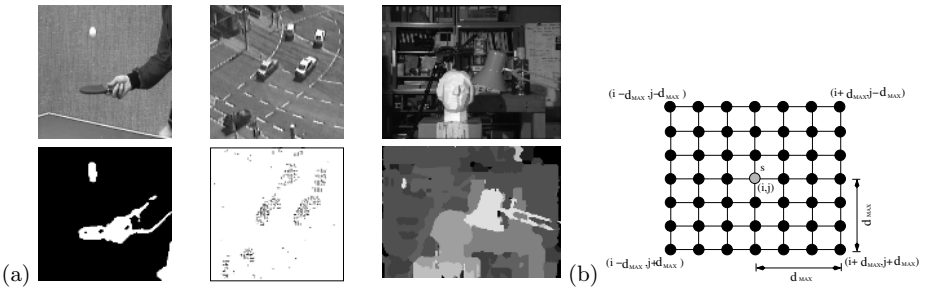


Fig. 1. (a) Motion detection, motion estimation, and stereovision label fields obtained after an ICM optimization. (b) The total number of possible displacement vector for each site $s \in S$ is $(2d_{max}+1)^2$.

paper, the system probability is made of the global energy function (here $U(x,y)$) and a temperature factor T. This probability function is similar to Boltzmann's probability function [9] which can be written as

$$P(x,y) = \frac{1}{\lambda}\exp\{-\frac{U(x,y)}{T}\}. \tag{10}$$

where λ is a normalization factor. The SA algorithm is presented in Table 1.

Table 1. Simulated annealing and ICM algorithms

1 $T \leftarrow T_{\text{MAX}}$
2 For each site $s \in S$
2a $P(x_s = e_i \mid y_s) = \frac{1}{\lambda}\exp\left\{\frac{-1}{T}U(e_i, y_s)\right\}, \quad \forall e_i \in \Gamma$
2b $x_s \leftarrow$ according to $P(x_s \mid y_s)$, randomly select $e_i \in \Gamma$
3 $T \leftarrow T *$cooling Rate
5 Repeat steps 2-3 until $T \leq T_{\text{MIN}}$

1 Initialize x
2 $x_s = \arg\min_{e_i \in \Gamma} U(e_i, y_s)\ \forall s \in S$
3 Repeat step 2 until x stabilizes

The major limitation with SA is the number of iterations it requires to reach the frozen ground state. This makes SA unsuitable to many applications for which time is an important factor. This explains the effort of certain researchers to find faster optimization procedures. One such optimization procedure is Besag's ICM algorithm [10]. Starting with an initial configuration $x^{[0]}$, ICM iteratively minimizes $U(x,y)$ in a deterministic manner by selecting, for every site $s \in S$, the label $e_i \in \Gamma$ that minimizes the energy function at that point. Since ICM isn't stochastic, it cannot exit local minima and thus, requires x to be initialized near the global minima. ICM algorithm is presented in Tab 1.

5 Graphics Hardware Architecture

Most graphics hardware are designed to fit the so-called *graphics processing pipeline* [16,17]. This pipeline is made of various stages which sequentially transform images and geometric input data into an output image stored in a section of graphics memory called the *framebuffer*. Part of the framebuffer (the front buffer) is meant to be visible on the display device. During the past few years, the major breakthrough in graphics hardware has been that the vertex processing and fragment processing stages have been made *programmable*. These two stages can now be programmed using C-like languages to process vertex and fragments in parallel. Because the GPU is an inherently parallel processing unit, mapping general computation problems to its unique architecture becomes very interesting [3].

The fragment processor is better suited for image processing problems than the vertex processor, simply because it is the only part of the graphics pipeline that has access to both input memory (texture memory) and output memory (the framebuffer). Let us mention that a fragment is a per-pixel data structure created at the rasterization stage and containing data such as color, texture coordinates and depth. A fragment is meant to update a unique location in the framebuffer. This location covers one or many pixels.

5.1 Markovian Segmentation on GPU

As one might expect, fragment programs (also called *fragment shader*) have some specificities as compared to ordinary C/C++ programs. The most important ones are the following:

1. a fragment program is made to process each fragment in parallel;
2. the *framebuffer* and the *depthbuffer* are the only memory a fragment program can write into;
3. the only data a fragment program can read is contained in the texture memory, in built-in variable or in user-defined variable. It cannot read the content of the framebuffer or the depthbuffer;
4. since a fragment program cannot read the framebuffer and since each fragment are processed in parallel, fragment programs cannot exchange information. GPUs do not provide its programs with access to general-purpose memory.

With such limitations, minimizing a global Markovian energy function such as Eq.(3) can be tricky. In fact, three main problems have to be overcome. Firstly, when performing a Markovian segmentation, fragment operations should be performed on every pixel of the input scene. As such, a $1:1$ mapping from the input pixels to the output buffer pixels is necessary. This is achieved by rendering a screen-aligned rectangle covering a *window* with exactly the same size as the input image. This generates exactly the right amount of fragments in the graphics pipeline such that a label estimated by a fragment program will be copied into one and only one pixel of the framebuffer. Such implementation

is illustrated in Table 2. Notice that a fragment program is launched over each pixel when the rectangle is rendered (line 4). In this way, the $\mathcal{N} \times \mathcal{M}$ label field x is estimated with the help of one CPU program and $\mathcal{N} \times \mathcal{M}$ fragment programs. In other words, the CPU program renders the scene and manages the texture memory while a fragment program minimizes the energy function $U(x_s, y_s)$ for each pixel.

Table 2. High level representation of an ICM hardware segmentation program. For an SA implementation, a temperature factor as well as a cooling rate shall be added. The first program (line 1 to 7) is the C/C++ CPU program loading the fragment program, rendering the scene and managing textures. The second program (line 1-2) is the fragment program launched on every pixel when the scene is rendered (line 4). Notice that images x and y are contained into texture memory.

1	Copy the input images into texture memory
2	Compile, link and load fragment shader (FS) on the GPU
3	Specify parameters of FS (β_{MD},β_{ME} or β_{S} for example)
4	Render a rectangle covering a window of size $\mathcal{N} \times \mathcal{M}$
5	Copy the framebuffer into texture memory
6	Repeat steps 4 and 5 until convergence
7	Copy the framebuffer into a C/C++ array if needed

1	$\hat{x}_s \leftarrow \arg\min_{x_s} U(x_s, y_s)$
2	$\text{framebuffer}_s \leftarrow \hat{x}_s$

The second problem comes from the fourth limitation. Since GPUs provide no general-purpose memory, one might wonder how the prior energy function V_{η_s} can be implemented on a fragment program since it depends on the neighboring labels x_t contained in the (write-only) framebuffer. As shown in Table 2, after rendering the scene, the CPU program copies the framebuffer into texture memory (line 5). In this way, the texture memory contains not only the input images, but also the label field x computed during the previous iteration. Thus, V_{η_s} is computed with labels iteratively updated and not sequentially updated as it is generally the case. Such strategy was already proposed by Besag [10] and successfully tested by other authors [12]. Notice that the iterative nature of ICM and SA is reproduced with multiple rendering of the rectangle (lines 4-5-6).

The last problem with shaders comes with their inability to generate random numbers such as needed by SA. As a workaround, we generate a random image that we copy in texture memory where the shader can access it.

5.2 Computer Vision on GPU

With the technique illustrated in table 2, performing motion detection, motion estimation and stereovision on a GPU is fairly straightforward. Since the shading languages available to write fragment programs (NVIDIA's *Cg* language in our case) are similar to C, the software programs can be reused almost directly.

The implementation of the three fragment shaders is conceptually very similar since they all minimize an energy function made of a likelihood term and a prior term. There is one exception though when stereovision requires a pre-filtering step ($(w * C)(s, x, y)$). We deal with this situation by pre-computing $C(s, x_s, y_s)$ in a 3D DSI table located in texture memory. This 3D table is then filtered with w after which the optimization procedure (SA, ICM or WTA) is launched.

6 Experimental Results

Results compare software and hardware implementations of the three applications we have discussed so far. The goal being to show how fast a segmentation program implemented on a GPU is compared to its implementation on a CPU. The software programs were implemented in C++ while NVIDIA's Cg language was used to implement the fragment shaders. All programs were executed on a conventional home PC equipped with a AMD Athlon 2.5 Ghz, 2.0 Gig of RAM and a NVIDIA 6800 GT graphics card. NVIDIA fp40 Cg profile was used in addition to the cgc compiler version 1.3.

Every results were made after varying some variables. In Fig. 2, the lattice size vary between 64×64 and 1024×1024, the number of disparities D_{MAX} between 4 and 32, d_{MAX} between 2 and 6, and the aggregation window size

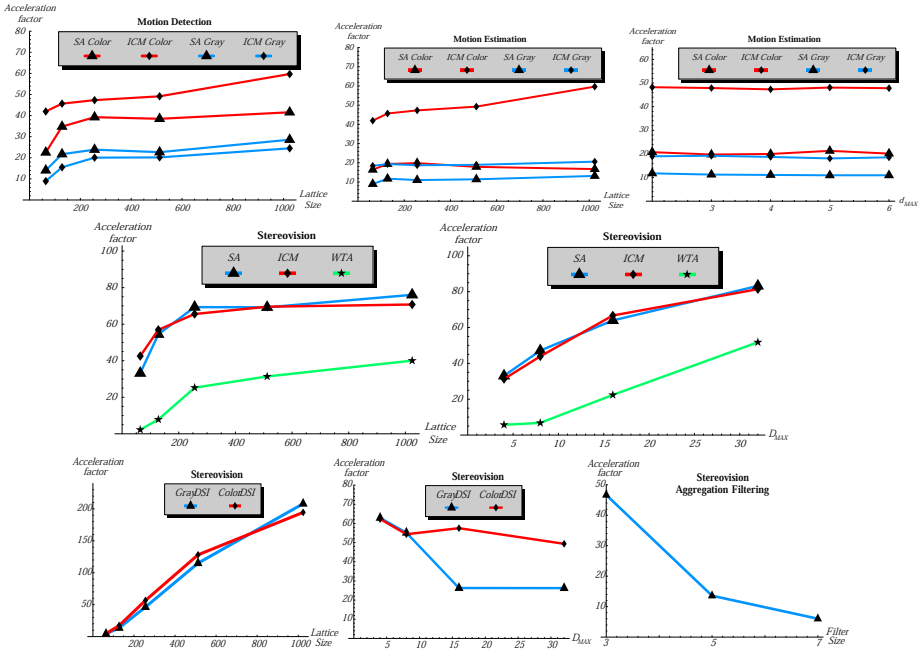


Fig. 2. Acceleration factor for the motion detection, motion estimation and stereovision programs

between 3×3 and 7×7. The number of iterations was set to 10 for ICM and to 500 for SA. Every results are expressed as an acceleration factor between the software and hardware execution time.

As shown in Fig 2, the hardware implementation is faster than its software counterpart by a factor between 10 and 100 approximately. Notice that the acceleration factor is generally more important for color sequences than for grayscale sequences. This is explained by the fact that the likelihood energy function W is more expensive to compute with RGB vectors than for grayscale values. Thus, distributing this extra load on a fragment processor results in a more appreciable acceleration factor.

For stereovision, we have tested the three tasks we have made reference to in Section 5.2, namely the computation of DSI, the aggregation filtering, and the optimization procedure (SA, ICM and WTA). As can be seen, the acceleration factor for ICM and SA is more important than for WTA. This can be explained by the fact that WTA is a very trivial and efficient algorithm (it converges in only one iteration). The computational load to distribute on the GPU is thus less important than for ICM and SA.

7 Conclusion

This paper shows how programmable graphics hardware can be used to performe typical energy-based segmentation applied to computer vision. Results show that the parallel abilities of GPUs significantly accelerate these applications (by a factor of 10 to 100 approximately) without requiring any specific skills in hardware programming. Such hardware implementation is usefull especially when the image size is large, when the number of labels is large or when the number of iteration is large.

References

1. Moreland K. and Angel E. The fft on a gpu. In *in proc. of Workshop on Graphics Hardware*, pages 112–119, 2003.
2. Kruger J. and Westermann R. Linear algebra operators for gpu implementation of numerical algorithms. *ACM Trans. Graph.*, 22(3):908–916, 2003.
3. http://www.gpgpu.org/.
4. R. Strzodka and M. Rumpf. Level set segmentation in graphics hardware. In *Proc. of ICIP*, 3, pages 1103–1106, 2001.
5. P. Bouthemy and P. Lalande. Motion detection in an image sequence using gibbs distributions. In *Proc. of ICASSP*, pages 1651–1654, 1989.
6. Konrad J. and Dubois E. Bayesian estimation of motion vector fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(9):910–927, 1992.
7. Scharstein D., Szeliski R., and Zabih R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proc. of the IEEE Workshop on Stereo and Multi-Baseline Vision*, 2001.
8. Geman S. and Geman D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6(6):721–741, 1984.

9. Kirkpatrick S., Gelatt C., and Vecchi M. Optimization by simulated annealing. *Science*, 220, 4598:671–680, 1983.
10. Besag J. On the statistical analysis of dirty pictures. *J. Roy. Stat. Soc.*, 48(3):259–302, 1986.
11. Chou P. and Brown C. The theory and practice of bayesian image labeling. In *Proc. of ICCV*, pages 185–210, 1990.
12. Dumontier C., Luthon F., and Charras J-P. Real-time dsp implementation for mfr-based video motion detection. *IEEE Trans. on Img. Proc.*, 8(10):1341–1347, 1999.
13. Nagel H-H. Image sequence evaluation: 30 years and still going strong. In *proc. of ICPR*, pages 1149–1158, 2000.
14. Mitiche A. and Bouthemy P. Computation and analysis of image motion: a synopsis of current problems and methods. *Int. J. Comput. Vision*, 19(1):29–55, 1996.
15. Black M. and Anandan P. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Comput. Vis. Image Underst.*, 63(1):75–104, 1996.
16. Randi J. Rost. *OpenGL Shading Language.* Addison-Wesley, 1st edition, 2004.
17. Tomas Akenine-Möller and Eric Haines. *Real-time Rendering.* AK Peters, 2e edition, 2002.

Efficient Hardware Architecture for EBCOT in JPEG 2000 Using a Feedback Loop from the Rate Controller to the Bit-Plane Coder

Grzegorz Pastuszak

Warsaw University of Technology, Institute of Radioelectronics,
Nowowiejska 15/19, 00-665 Warsaw, Poland
G.Pastuszak@ire.pw.edu.pl

Abstract. At the low compression ratio, the EBCOT engine of the JPEG 2000 encoder does not have to process all input data to achieve an optimal codestream in the sense of the rate-distortion criteria. This property is exploited in the architecture presented in this paper to allow higher throughputs of the JPEG 2000 encoder. An impact of the code block size and the internal FIFO size on the resultant speed is considered. The architecture is described in VHDL and synthesized for commercial FPGA technology. Simulation results show that at low compression ratios and for FPGA Stratix II devices, the single engine can support HDTV standards.

1 Introduction

The newest compression standards allow ever-higher compression ratio and support new functionality, although their computational complexity is still increasing. In image compression, JPEG 2000 [1], [2] incorporates some sophisticated algorithms, which require efficient implementation methods to shorten execution time. Embedded Block Coding with Optimized Truncation (EBCOT) is central to the standard and is a bottleneck of the codec. Usage of hardware acceleration makes it possible to obtain high throughputs.

There are some architectures and optimization methods for EBCOT presented in literature [3]-[7]. Also, some commercial solutions are available one the market, however, details are not published. Most implementation works for EBCOT have focused on the Bit-plane coder (BPC) and assumed that the Context Adaptive Binary Arithmetic Coder (CABAC) can process at most one binary symbol per clock cycle. We proved that it is possible to build efficient EBCOT architectures able to code two or three symbols per clock cycle [8]. Lossy compression gives an opportunity to shorten execution time since a number of input data do not contribute to the final JPEG 2000 codestream. There are a variety of design strategies exploiting this property. Some of them were proposed in [9], [10]. The main concept behind them is to skip less significant bit-planes of input coefficients based on rate-distortion criteria. This paper applies these approaches to the architecture able to code two symbols per clock cycle. Moreover, an impact of the code block size and the internal FIFO size on the resultant speed is considered.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 604 – 611, 2005.

The rest of the paper is organized as follows: Section 2 reviews the EBCOT algorithm. Hardware acceleration methods and proposed design strategies for skipping of bit-planes are presented in Section 3. The architecture design is illustrated in Section 4. Section 5 gives evaluation results. Finally, Section 6 concludes the work.

2 EBCOT Algorithm

In the JPEG 2000 compression schema, input images undergo in turn: color transformation, wavelet transformation, and quantization. Each of these stages can be skipped depending on desired coding options. Quantized indices are grouped into rectangular structures (so-called code-blocks) and entropy-coded using the EBCOT algorithm.

2.1 Embedded Block Coding

The embedded block coder is known also as Tier-1 coder. The bit-plane coder is the first stage of the EBCOT algorithm. The BPC generates context-symbol pairs on the basis of quantization indices grouped in code-blocks. Input data are read in the sign-magnitude format and analyzed bit-plane wise starting from the most significant bit-plane (MSB) with a non-zero element to the least significant bit-plane (LSB). Each bit-plane is scanned in three coding passes called significance propagation, magnitude refinement, and cleanup.

Each pass provides a variable quality contribution to the reconstructed image. For the sake of the rate control algorithm, such an improvement should be calculated as a reduction in distortion, which may be obtained by summing reductions associated with each processed coefficient. For a single coefficient, the reduction in distortion can be calculated from bits located under the currently scanned bit-plane.

The context adaptive binary arithmetic coder (CABAC) is the second stage of the EBCOT algorithm. The CABAC module reads context-symbol pairs from the bit-plane coder and codes them into separate bit streams for each code-block. The CABAC contains the finite state machine that keep probability model for each context. The model identifies a binary value of the most probable symbol (MPS) and keeps an index pointing probability estimate of the least probable symbol (LPS).

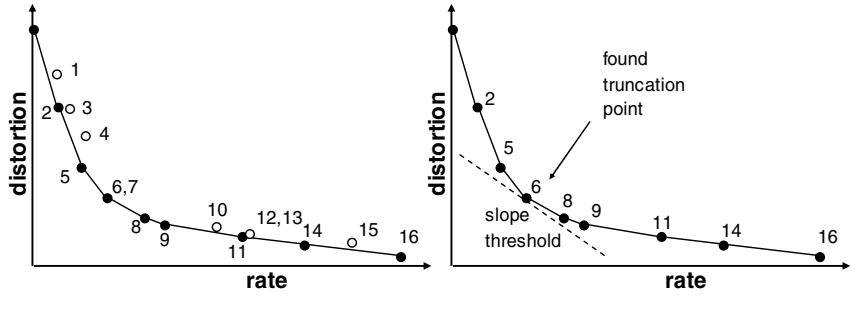


Fig. 1. Convex hull analysis and finding truncation point for a code-block

The main coding routine of the CABAC is based on the interval subdivision method of Shannon and Elias. The interval is described by its length and base. Successive renormalizations release bytes from MSB position of the base, incrementing the byte counter. Discrete truncation rates for each coding pass may be estimated on the basis of this counter increased by a small number (from 1 to 5) calculated from the internal variables. When the last pass is completed, the truncation length is equal to the number of released bytes.

2.2 Rate Control

The JPEG 2000 coder can employ Discrete Lagrange Optimization to achieve the target rate with a high accuracy. Inputs to the Discrete Lagrange Optimization are allowable truncation points described by reductions in distortion and rates, which are obtained from the BPC and the CABAC. The method finds the best truncation points for each code block to minimize the distortion of the reconstructed image subject to the target rate.

For each code block, the submitted points should form a convex hull to provide monotonic dependency of their slopes $\Delta D/\Delta R$. In order to meet this condition, some points have to be removed, as depicted in Fig. 1. The optimization procedure determines a global threshold with reference to the slopes. Next, a codestream for each code-block is truncated at the rate corresponding to the point having the smallest slope but greater than the global threshold. The final codestream including such truncated codestreams from each code block is optimal in the sense of the rate-distortion criteria. The JPEG 2000 encoder has to find the threshold that provides matching between the target and achieved rates.

3 Hardware Acceleration Methods

3.1 Embedded Block Coding

As each code-block is entropy-coded independently, the opportunity for parallel processing arises by using several block-coding engines. Speedup techniques for the BPC employ simultaneous scanning of one or more columns in a stripe and skipping no-operation samples. It reduces significantly local discontinuities of the output stream produced by the BPC unit. Another technique uses a FIFO buffer between the BPC and the CABAC. The reduction of time intervals, when input data for the CABAC are not available, depends on the FIFO size and the difference in speed between both main modules of the entropy coder. Most notably, the faster generation mitigates requirements for the capacity of the FIFO.

The main limitation on the throughput of EBCOT arises from the casual dependencies existing in the CABAC algorithm. Using a pipeline arrangement and parallel symbol encoding can increase the throughput.

3.2 Feedback Loop from the Rate Controller to the Bit-Plane Coder

Lossy compression gives an opportunity to shorten execution time since not all passes from a given code block contribute to the final JPEG 2000 codestream. To benefit

from this feature, the BPC should terminate processing of the code-block at the last pass included in the final codestream. In practice, this condition is difficult to evaluate since the optimal selection of last passes depends on the rate-distortion relations between all code-blocks in the image, as described in subsection 2.2. Hence, optimal truncation points can be found after all the code-blocks have been processed.

Final truncation points have rate-distortion slopes ($\Delta D/\Delta R$) greater than the threshold found in the Discrete Lagrange Optimization. An estimation of the threshold in parallel with processing of code-blocks allows the BPC to skip passes violating this condition. In [9], the estimation is accomplished by assigning the target rate to code-blocks processed so far. Thus, the value of the threshold increases during coding until the final value is found. Underestimation of the threshold adversely affects the efficiency of the early termination, e.g. the speedup is not as great as it would be. Owing to the temporal correlation in Motion JPEG 2000, the threshold can be taken from the preceding frame and modified with accordance to an adaptation rule [10].

An additional problem in embedding the feedback loop from the rate controller arises from the locally non-monotonic dependence between slopes corresponding to successive passes. This makes the correct detection of the termination condition difficult. For example, it may happen that the slope for a pass can fall below the temporal threshold validating the termination condition, but the slope of the next pass is greater and even merging these passes in the convex-hull analyzer makes the condition false. In this case, the termination would be performed too early.

The underestimation of the threshold cancels the negative impact of the too-early termination on the quality. The strength of these two factors should be balanced in order to prevent losses in quality and to minimize over-coding. This can involve additional conditions for the termination regarding the growth of the codestream length, the inclusion of zero-in-length cleanup passes, and the number of passes included in the collocated code-block in the preceding frame.

In hardware framework, the termination decision can be taken for a pass when both the reduction in distortion and the growth in the codestream length are known. The latter is determined with a delay caused by buffering (FIFO) and pipelining between the BPC and the rate estimator following the CABAC. As a consequence of the fact that the BPC continues processing until the termination, the delay has a similar effect on the coding time as the underestimation of the threshold. Varying the size of the FIFO changes the strength of this effect.

4 Architecture

4.1 Block-Coding Path

The BPC applies a pipeline arrangement and employs six memory modules to buffer quantized indices (2x1024x13 bits and 4x512x13 bits) and two memories to keep state variables (2x512 bits). There is a FIFO buffer as the last stage of the BPC. The bit-plane analysing method produces data at an average rate that outperforms by far the throughput of the CABAC able to code two context-symbol pairs per clock cycle. In particular, four columns in a stripe (16 bits) are scanned in one clock cycle. When there are more symbols to encode, additional clock cycles are inserted. To calculate

the reduction in distortion for each pass, bits from three bit-planes located just under the currently scanned bit-plane are read in parallel. If a coefficient is or becomes significant, its bits from all these bit-planes are mapped onto a value in the square error domain, and the result is added to a mean-square error (MSE) accumulator. At the end of each pass, the accumulator is flushed out.

The CABAC applies five pipeline stages. It is designed to process two context-symbol pairs per clock cycle [8]. The rate estimator, following the CABAC, calculates truncation rates on the basis of states of internal registers of the CABAC and generated code-bytes. The applied estimation of truncation rates is close to optimal because it discards code-bytes that are not necessary to decode correctly the last pass of a code-block.

4.2 Feedback Loop from the Rate Controller

The truncation rates along with the quality reduction, expressed as MSE, are forwarded from the block-coding path to the rate control one. The latter consists of a convex-hull analyzer, a feedback-threshold estimator, and the main rate controller.

The convex-hull analyzer embeds two small FIFO buffers to collect and adjust truncation rates and corresponding MSE reductions. A finite state machine (FSM) controls calculations of slopes and removes truncation points violating conditions on the convex hull. The convex-hull block incorporates a subcircuit converting inputs to their logarithmic representation. This removes the need to use the multiplication and division units and keeps the high dynamic range of slopes at 16 bits.

Positively classified truncation points are stored in a double-port memory interfacing the convex-hull analyzer with the feedback-threshold estimator. Both blocks access to two separate address spaces through their own ports. Exchange of address spaces allows communication. Hence, the blocks can operate simultaneously on successive code-blocks.

The feedback-threshold estimator provides the temporal threshold to the convex-hull analyzer, which in turn compares it with slopes calculated for successive truncation points of the currently-processed code-block. If a slope is less than the threshold, a termination signal is activated driving the BPC to finish coding at the end of the current pass. The feedback-threshold estimator handles a slope table accumulating rates. Each truncation point adds its rate to an accumulator addressed by the slope of this point. In the designed architecture, the threshold and slopes occupy 16 bits. To save hardware resources, eight most significant bits address the table. After updating the table by truncation points from a code-block, the temporal threshold is determined. The accumulators are read starting from the highest address. Their rates are accumulated in a global accumulator until its value is less than the total rate. The threshold is equal to the address of the table accumulator causing the violation of this condition. The architecture incorporates two slope tables accessed alternately on the image basis between the feedback threshold estimator and the Tier-2 coder.

All truncation points and code-bytes are forwarded to the external memory. When all code-blocks are coded, the Tier-2 of the JPEG 2000 encoder is activated. In particular, accurate truncation rates are determined, packet headers are coded, and final codestream is produced. Truncation rates are found on the basis of the accurate threshold by comparing it with the slope of each truncation point within each code-

block. Truncation rates are taken from the least significant points having slope greater than the threshold. The accurate threshold is determined in two steps. High order bits are obtained by reusing the slope table built by the feedback threshold estimator. Next, the table is reinitialized by truncation points (stored in the external memory) whose slopes have high-order bits equal to those of the threshold. Less significant bits are retrieved in the similar manner as higher ones.

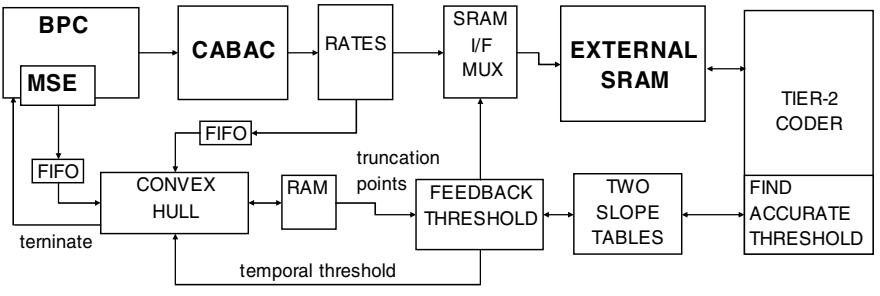


Fig. 2. Block diagram of the EBCOT engine

5 Implementation Results

The designed architecture of the EBCOT engine has been described in VHDL and verified against software reference model (JJ2000 version 5.1). Synthesis for the FPGA technology has been performed. The Tier-1 engine consumes 10 K Logic Elements and can operate at 120 MHz working frequency for FPGA Stratix-II devices. It enables the encoder to process about 40 million samples in the lossless mode (RCT, 5x3 wavelet filter core, no quantization). In the lossy mode, the throughput is higher and depends on applied quantization steps and the efficiency of the feedback loop from the rate controller.

Evaluations have been conducted for a set of images. Tables 1 and 2 show the number of clock cycles utilized to code the Baboon image by the block-coding engine enhanced or not by the feedback loop. The results for the Baboon image are presented in this paper since it involves the greatest deal of computations compared to other images. Evaluation conditions have been as follows: 512x512 grayscale image, ICT, 9x7 wavelet transform, two decomposition levels, quantization adjusted to the L2 norm of the wavelet filter. The tables show that the compression ratio, the size of the FIFO between the BPC and the CABAC, and the code-block size have an impact on the processing speed. In particular, lower compression ratio allows higher throughputs. Without the feedback loop, the number of clock cycles decreases with increasing the FIFO size. As discussed in subsection 3.2, this rule does not hold when exploiting the feedback loop. In this case, the number of clock cycles attains a minimum at some sizes of the FIFO, and it depends on the compression ratio, the code-block size, and the content of an image.

The tables compare two strategies of the threshold estimation. The first one approximates the temporal threshold by assignment of the total rate to code-blocks processed so far (increasing threshold). The second one applies the final threshold to all code-blocks), as this threshold would be taken from a preceding frame in a sequence (threshold from preceding frame). In this case, the reduction of clock cycles is the largest. When the threshold is underestimated, the savings in processing time are less. In some cases, the too-early termination can occur slightly deteriorating the quality of the reconstructed image. Nevertheless, these quality losses, if present, are very small (changes in PSNR are less than 0.05 dB) and can be neglected.

Table 1. Number of clock cycles necessary to code the Baboon image in the Tier-1 part for Code-block size 16x16

FIFO size x 32 bits		4	8	16	32	64
No feedback loop		807024	773855	738469	707850	707818
Increasing threshold	2 bpp	768179	739773	710901	698120	707124
	1 bpp	650985	633424	616659	615391	672120
	0.5 bpp	548867	536285	526335	529853	604490
Threshold from preceding frame	2 bpp	580200	568267	556982	560655	664117
	1 bpp	445103	434030	430286	434136	530044
	0.5 bpp	386522	378511	378001	386473	474921

Table 2. Number of clock cycles necessary to code the Baboon image in the Tier-1 part for Code-block size 64x64

FIFO size x 32 bits		32	64	128	256	512	1024
No feedback loop		848462	824180	796203	762422	736206	736157
Increasing threshold	2 bpp	791621	770341	753618	731745	717189	736154
	1 bpp	661254	644060	637568	617714	626650	686898
	0.5 bpp	540362	525738	516151	510806	520815	592612
Threshold from preceding frame	2 bpp	549247	541842	540840	530591	550585	664009
	1 bpp	386454	377120	367823	361692	368788	468429
	0.5 bpp	326829	316069	306247	296730	314427	399750

6 Conclusions

The architecture of the EBCOT with the feedback loop from the rate controller has been designed. It has been described in VHDL, verified, and synthesized for FPGA Stratix-II devices. The engine can operate at 120 MHz working frequency and can support HDTV standards at the low compression ratio.

Exploiting the feedback threshold from the rate controller improves the throughput of the EBCOT engine. The optimal selection of the FIFO size depends on the code-block size, the compression ratio. The temporal prediction of the threshold from the preceding frame gives the best speedup.

Acknowledgement

The work presented was developed within activities of VISNET, the European Network of Excellence, (http://www.visnet-noe.org), founded under the European Commission IST 6FP programme.

References

1. ISO/IEC 15444-1, Information technology - JPEG 2000 image coding system - Part I: Core Coding System, 2000
2. Taubman, D. S., Marcellin, M. W., JPEG2000: Image Compression Fundamentals, Standard and Practice. Norwell, MA: Kluwer, (2002)
3. Hsiao, Y.-T., Lin, H.-D., Lee, K.-B., Jen, C.-W.: High Speed Memory Saving Architecture for the Embedded Block Coding in JPEG 2000. (2002).
4. Andra, K., Chakrabarti, C., Acharya, T.: A high performance JPEG2000 architecture, IEEE Trans. Circuits and Systems for Video Technology, vol. 13, no. 3, pp. 209–218, (2003).
5. Lian, C.-J., Chen, K.-F., Chen, H.-H., Chen, L.-G.: Analysis and architecture design of block-coding engine for EBCOT in JPEG 2000, IEEE Trans. Circuits and Systems for Video Technology, vol. 13, no. 3, pp. 219–230, (2003).
6. Li, Y., Aly, R.E., Wilson, B., Bayoumi, M.A., Analysis and Enhancement for EBCOT in high speed JPEG 2000 Architectures, The 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002., Volume: 2. (2002)
7. Fang, H.-C., Wang, T.-C., Lian, C.-J., Chang, T.-H., Chen, L.-G.: High Speed Memory Efficient EBCOT Architecture for JPEG2000, Proceedings of the 2003 International Symposium on Circuits and Systems, Vol. 2, pp 736-739, May 2003.
8. 8 Pastuszak, G.: A High-Performance architecture for arithmetic Coder in JPEG2000, ICME'04, Taipei, Taiwan, 2004.
9. Chang, T.-H., Lian, C.-J., Chen, H.-H., Chang, J.-Y., Chen, L.-G.: Effective Hardware-Oriented Technique For The Rate Control Of JPEG2000 Encoding. International Symposium on Circuits and Systems ISCAS 2003, Bangkok, Thailand. (2003)
10. Yu, W., Fritts, J., Fangting S.: Efficient Rate Control for Motion JPEG2000, Data Compression Conference (DCC 2004), Snowbird, UT, USA (2004)

Incorporating Geometry Information with Weak Classifiers for Improved Generic Visual Categorization

Gabriela Csurka, Jutta Willamowski, Christopher R. Dance, and Florent Perronnin

Xerox Research Centre Europe,
6 Rue de Maupertuis, 38240 Meylan, France
{gsurka, willamow, cdance, fperronn}@xeroxlabs.com

Abstract. In this paper[1], we improve the performance of a generic visual categorizer based on the "bag of keypatches" approach using geometric information. More precisely, we consider a large number of simple geometrical relationships between interest points based on the scale, orientation or closeness. Each relationship leads to a weak classifier. The boosting approach is used to select from this multitude of classifiers (several millions in our case) and to combine them effectively with the original classifier. Results are shown on a new challenging 10 class dataset.

1 Introduction

The proliferation of digital imaging sensors in mobile phones and consumer-level cameras is producing a growing number of large digital image collections and increasing the pervasiveness of images on the web and in other documents. To search and manage such collections it is useful to have access to high-level information about objects contained in the images. We are interested in recognizing several objects or image categories within a multi-class categorization system, but not in the localization of the objects unnecessary for most applications involving tagging and search. Therefore, in this paper we propose a system which is sufficiently generic to cope with many object types simultaneously and which can readily be extended to new categories. It can handle variations in view, background clutter, lighting and occlusion as well as intra-class variations.

The main novelty is to exploit a boosting approach based on interest points and simple geometrical relationships (similar scales, similar orientation, closeness) between them. We chose to adopt the boosting approach because there are many possible geometric relationships and boosting offers an effective way to select from this multitude of possible features. It was used with success in [11] to detect the presence of bikes, persons, cars or airplanes against background. However their approach differs from ours as they do not include any geometry and consider every appearance descriptor (over 2 million for our data set) without considering a vocabulary, which is impractical if geometric combinations of such descriptors are to be exploited.

The main advantage of our approach is that geometric constraints are introduced as weak conditions in contrast to others such as [4,7], where due to relatively strong

[1] This work was funded by the EU project LAVA (IST-2001-34405).

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 612–620, 2005.

geometrical (shape) constraints the methods requires the alignment and segregation of different views of objects in the dataset.

Several categorization approaches have recently been developed that are based on image segmentation [1,8,12,2], rather than the interest point descriptors exploited here. Some of these works attempt to solve the more difficult problem of labeling several regions per image. In [1] geometry has been included through MRF models of neighboring relations between segmented regions. In contrast we prefer to take a discriminative classifier approach in order to optimize overall accuracy.

The remainder of this paper is organized as follows: Section 2 describes the original SVM approach; in Section 3 we introduce an alternative based on the boosting framework; in section 4 we then describe how to incorporate weak geometry in the boosting approach and we finally present preliminary experimental results in section 4.

2 The Original Approach

We describe very briefly the original visual categorization approach introduced in [3]. The main steps of our method as applied to labeling a previously unseen image are as follows. We detect image patches and assign each of them them to one of a set of predetermined clusters (a vocabulary), on the basis of their appearance descriptors[2]. We then apply one classifier (SVM) per visual category with a one-against-all approach to handle the multiple visual categories.

The extracted descriptors of image patches should be invariant to the variations that are irrelevant to the categorization task (image transformations, lighting variations and occlusions) but rich enough to carry all necessary information to be discriminative at the category level. We used Lowe's SIFT approach [9] to detect and describe image patches. This produces scale-invariant circular patches that are associated with 128-dimensional feature vectors of Gaussian derivatives. While in [3] we used affine invariant elliptical patches [10], similar performance was obtained with circular patches. Moreover, the use of circular patches makes it simpler to deal with geometric issues.

The visual vocabulary was constructed using the k-means algorithm applied to a completely independent set of images with over 10,000 patches. We are not interested in a correct clustering in the sense of feature distributions, but rather in an accurate categorization. Therefore, to overcome the initialization dependence of k-means, we run it several times with different initial cluster centers and select the final clustering giving the highest categorization accuracy using an SVM classifier (without any geometric properties) on a subset of the dataset.

For categorization we use the SVM, which finds the hyperplane that separates two-class data with maximal margin [17]. The SVM decision function can be expressed as $f(\mathbf{x}) = \text{sign}(\sum_i y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b)$, where $\mathbf{x}_i$ are the training features from data space and $y_i \in \{-1, 1\}$ is the label of $\mathbf{x}_i$. In this paper, the input features $\mathbf{x}_i$ are the binned histograms formed by the number of occurrences of keypatches in the input image. K is a kernel function corresponding to an inner product between two transformed feature vectors, usually in a high and possibly infinite dimensional space. In our experiences

[2] In this paper, we will refer to patches assigned in this way as *keypatches* instead of *keypoints*.

we used a linear kernel, which is the dot product of $\mathbf{x}$ and $\mathbf{x}_i$. The parameters α_i are zero for most i, the sum is taken only over a selected set of $\mathbf{x}_i$ known as support vectors.

3 The Boosting Approach

An alternative to the SVM classifier is the boosting approach. Here we exploit the generalized version of the AdaBoost algorithm described in [15] and improved by Rätsch *et al* [14] by adding soft margins. Boosting is a method of finding an accurate classifier H by combining M simpler classifiers h_m:

$$H(\mathbf{x}) = \Big(\sum_{m=1}^{M} \alpha_m h_m(\mathbf{x}) \Big) / \Big(\sum_{m=1}^{M} \alpha_m \Big)$$

Each simpler classifier $h_m(\mathbf{x}) \in [-1, 1]$ needs only to be moderately accurate and is therefore known as a *weak classifier*. They are chosen from a classifier space to maximize correlation[3] of the predictions and labels $r_m = \sum_i D^m(i) h_m(\mathbf{x}_i) y_i$, where $D^m(i)$ is a set of weights (distribution) over the training set. At each step the weigths are updated by increasing the weights of the incorrectly predicted training examples:

$$D^{m+1}(i) = D^m(i) \exp\{-\alpha_t y_i h_m(\mathbf{x}_i)\} / Z_m$$

where $\alpha_t = \frac{1}{2} \log \frac{1+r_m}{1-r_m}$ and Z_m is a normalization constant, such that $\sum_i D^{m+1}(i) = 1$. In the case of soft margins, α_t and $D^{m+1}(i)$ depend also on a regularization term which takes into consideration the predictions and weights produced by the previous steps in order to eliminate the influence of outliers [14].

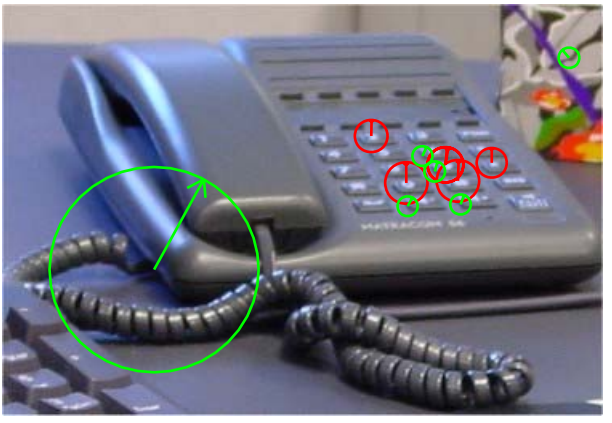
To define the weak classifiers we consider the same inputs as for the SVM, i.e. the binned histograms $\mathbf{x}_i$. The simplest keypatch-based weak classifier $h^{k,T}$ counts the number of patches whose SIFT features belong to cluster k, which is equivalent to compare $\mathbf{x}_i^k$ to the threshold T. If this number is at least T, then the classifier output is 1, otherwise 0. We may build similar weak classifiers $h^{k,l,T}$ from a pair of keypatch types k, l. If at least T keypatches of both types are observed, then the classifier output is 1.

In practice we select weak classifiers by searching over a predefined set of thresholds such as $\{1, 5, 10\}$. The opposite weak classifier $h^{k,\bar{T}}$ can also be defined by inversing the inequality. Four such definitions are possible for pairs of keypatches $h^{kl,T}$, $h^{kl,\bar{T}}$, $h^{kl,T\bar{T}}$ and $h^{kl,\bar{T}T}$. In practice, we search over the full set of different possibilities when working with weak classifiers and refer to them collectively as h^k and h^{kl}. Obviously, it would be possible to further extend the definition for pairs to applying a different threshold to each keypatch type. In practice, we avoid this as it results in a prohibitively large number of possible weak classifiers.

4 Incorporating Weak Geometric Information

In this section we describe some of the possibilities to construct weak geometrical constraints on image patches. As input, we assume each patch i in a query image has been

[3] This is equivalent to minimizing the training error equal to $(1 - r_m)/2$.



$h_{\sigma}^{g,5}$, $h_{\sigma\theta}^{g,4}$, $h_{\sigma}^{rg,2}$, $h_{\theta}^{rg,5}$, $h_{\sigma\theta}^{rg,2}$, $h_{g\cap r}^{1}$ and $h_{g\in r}^{1} = 1$
$h^{r,6}$, $h_{\theta}^{g,6}$, $h^{rg,6}$, $h_{\sigma=}^{rg,1}$, $h_{\theta=}^{rg,1}$, $h_{\sigma\theta=}^{1}$ and $h_{r\subset r}^{1} = 0$

Fig. 1. Examples of weak classifiers on a typical image for keypatches of type r, g (red or green). For clarity, only the patches of type r and g are shown. In these examples, the threshold T on which the weak classifiers depend has been chosen as large as possible for output 1 (first row) and as small as possible for output 0 (second row).

labeled according to its appearance, via the index of the cluster centre k_i to which it is assigned. We associate to each patch its orientation θ_i and a ball (circular patch) B_i which has position p_i and scale σ_i.

A simple way to incorporate geometrical information in weak classifiers depending on one keypatch is to threshold the number of interest points belonging to a cluster k and having a particular *orientation*. A large number of different orientations are produced by interest point detectors. Therefore we exploit a coarse quantization of the orientations into eight bins. Two keypatches are considered to have the same orientation if they fall into the same bin. This does not constitute exact orientation invariance, as a small rotation could cause two keypatches in one bin to move to different bins. However, this approach is more efficient than directly measuring and thresholding the difference in orientations $\|\theta_i - \theta_j\|$ between pairs of keypatches.

Likewise, we define sets of weak classifiers that count the number of keypatches with the same *scale* and a set that count patches with both the same *scale and orientation*. The scale bins are selected with logarithmic spacing, in order to approximate scale invariance. Collectively[4] these classifiers are denoted by $h_{\theta}^{k}, h_{\sigma}^{k}, h_{\sigma,\theta}^{k}$.

Another way to incorporate geometry is to count the *number of interest points in the ball* around a keypatch of a given type. This count is made irrespective of the type of keypatches in the ball. As with the other weak classifiers, this property is invariant to shift, scaling and rotation. In a given image, there may be multiple keypatches of a given type containing different numbers of points. We define h_{N}^{k} in terms of the keypatch of type k with the maximum number of points in its ball.

[4] Considering similar threshold reversals as for h^{k} and h^{kl}, e.g. $h_{\theta}^{k,T}$ and $h_{\theta}^{k,\bar{T}}$.

Taking two types of keypatches k and l into consideration, there are more ways to introduce geometry. Classifiers based on common scale or orientation can be extended in two obvious ways. Firstly we can require that the patches of type k and those of type l have *identical* scale and/or orientation, giving $h^{kl}_{\sigma=}, h^{kl}_{\theta=}, h^{kl}_{\sigma\theta=}$. Alternatively we can allow each type to have their own independent scales or orientations, giving $h^{kl}_{\sigma}, h^{kl}_{\theta}, h^{kl}_{\sigma\theta}$. The latter corresponds to a Boolean combination of single point classifiers h^k_σ and h^l_θ.

A weak classifier h^{kl}_N can be constructed similarly to h^k_N that checks for the existence of a pair of interest points labeled k, l such that both of them have at least T interest points inside their balls.

We additionally consider five other ways of exploiting the position information associated with patches:

- $h_{k\in l}$ tests if there are at least T keypatches labeled l which contain an interest point labeled k within their ball.
- $h_{k\subset l}$ tests if there are at least T keypatches labeled l whose balls contain the whole ball of an interest point labeled k.
- $h_{k\cap l}$ tests if there are at least T keypatches labeled l whose balls intersect with the ball of at least one interest point labeled k.
- $h_{k\propto l}$ tests if there are at least T keypatches labeled l such that their closest neighboring interest points in the image are labeled k.
- $h_{k\in\aleph^N_l}$ tests if there are at least T keypatch labeled l such that there exist a keypatch labeled k among its N closest neighbors.

The set of weak classifiers we considered is summarized in Table 1 and Figure 1 illustrates some of them on a concrete example. Of course there are a lot of other possibilities that can be further experimented.

Table 1. Complete list of weak classifiers investigated. $\big|A\big|$ denotes the cardinality of the set A. $p \propto q$ indicates that p is the closest point to q and $\aleph^N_{p_j}$ is the set of the N closest neighbors of p_i

h	$h=1$ if this quantity $\geq T$	h	$h=1$ if this quantity $\geq T$
$h^{k,T}_{\sigma}$	$\max_{\sigma}\big\|\{i: k_i=k, \sigma_i=\sigma\}\big\|$	$h^{k,T}_{\sigma\theta}$	$\max_{\sigma,\theta}\big\|\{i: k_i=k, \sigma_i=\sigma, \theta_i=\theta\}\big\|$
$h^{k,l,T}_{\sigma}$	$\min_{u\in\{k,l\}}\max_{\sigma}\big\|\{i: k_i=u, \sigma_i=\sigma\}\big\|$	$h^{kl,T}_{\sigma\theta}$	$\min_{u\in\{k,l\}}\max_{\sigma,\theta}\big\|\{i: k_i=u, \sigma_i=\sigma, \theta_i=\theta\}\big\|$
$h^{k,T}_{\theta}$	$\max_{\theta}\big\|\{i: k_i=k, \theta_i=\theta\}\big\|$	$h^{kl,T}_{\sigma\theta=}$	$\max_{\sigma,\theta}\min_{u\in\{k,l\}}\big\|\{i: k_i=u, \sigma_i=\sigma, \theta_i=\theta\}\big\|$
$h^{kl,T}_{\theta}$	$\min_{u\in\{k,l\}}\max_{\theta}\big\|\{i: k_i=u, \theta_i=\theta\}\big\|$	$h^{T}_{k\in l}$	$\big\|\{j: k_j=l, \exists k_i=k, p_i\in B_j\}\big\|$
$h^{kl,T}_{\sigma=}$	$\max_{\sigma}\min_{u\in\{k,l\}}\big\|\{i: k_i=u, \sigma_i=\sigma\}\big\|$	$h^{T}_{k\subset l}$	$\big\|\{j: k_j=l, \exists k_i=k, B_i\subset B_j\}\big\|$
$h^{kl,T}_{\theta=}$	$\max_{\theta}\min_{u\in\{k,l\}}\big\|\{i: k_i=u, \theta_i=\theta\}\big\|$	$h^{T}_{k\cap l}$	$\big\|\{j: k_i=l, \exists k_i=k, B_i\cap B_j\neq\emptyset\}\big\|$
$h^{k,T}_{B}$	$\max_{i}\big\|\{j: k_i=k, p_j\in B_i\}\big\|$	$h^{T}_{k\propto l}$	$\big\|\{j: k_j=l, \exists k_i=k, p_i\propto p_j\}\big\|$
$h^{kl,T}_{B}$	$\max_{i}\min_{u\in\{k,l\}}\big\|\{j: k_i=u, p_j\in B_i\}\big\|$	$h^{T}_{k\in\aleph^N_l}$	$\big\|\{j: k_j=l, \exists k_i=k, p_i\in\aleph^N_{p_j}\}\big\|$

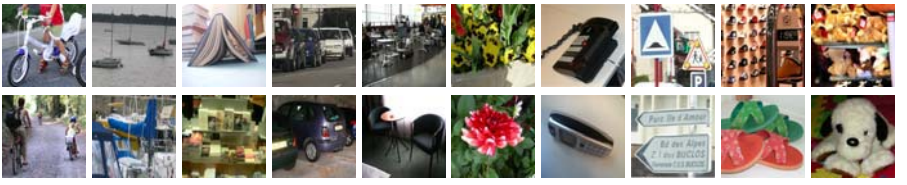


Fig. 2. Examples from our 10 class dataset

Table 2. Correct classification rates for: boosting without geometry ($h_k, h_{k,l}$); SVM with a linear kernel; boosting all types of weak classifiers h_{all} and boosting SVM with all type of weak classifiers (SVM_{all}). The standard error on the correct rate for each category is about 0.4% of the mean over the folds and is 0.2% for the overall mean.

classes	bikes	boats	books	cars	chairs	flowers	phones	road signs	shoes	soft toys	mean
h_k	61.7	74.5	67.0	55.6	50.7	82.5	67.6	61.4	73.9	68.9	**66.4**
$h_{k,l}$	64.6	76.1	68.5	61.0	50.7	84.6	69.6	64.8	76.6	69.2	**68.6**
SVM	69.2	79.3	70.3	72.1	58.8	86.7	70.4	69.0	86.3	79.2	**74.1**
h_{all}	70.0	73.8	68.2	64.1	57.4	82.9	68.0	61.9	75.2	76.2	**69.8**
SVM_{all}	74.6	81.8	78.2	77.5	65.2	89.6	76.0	76.2	83.8	83.8	**78.7**

5 Results

In our experiments we used a dataset of 3715 images from 10 categories[5]. The database can be downloaded from ftp://ftp.xrce.xerox.com/pub/ftp-ipc. Figures 2 shows some images from the database.

We experimented with several types of classifiers. In all cases we worked with a vocabulary of 1,000 keypatch types. This was selected as beeing a good trade-off between computational efficiency and classification accuracy.

First we compared directly the boosting approach with the SVM. The first three rows of Table 2 show the correct classification rate for each class: an image I of category i was considered as correctly classified if the output of the classifier $H_i(I) > H_j(I), \forall j \neq i$. All results are means of a 5-fold cross validation scheme. We can see that the SVM outperforms the Boosting approach.

Willing to incorporate weak geometry, we first preselected[6] from each type of geometry presented in Section 4 a certain number ($M = 200$) of weak classifiers that performed the best when only considering this type of weak hypotheses (e.g. $h^{kl}_{\sigma\theta}$). This preselection step allow us also to investigate how well each type of weak classifier combined trought a boosting framework perform independently (see Table 3).

[5] The number of images per class were: bikes(243), boats(439), books(272), cars(315), chairs(346), flowers(242), phones(250), road signs(211), shoes(525) and soft toys(262).

[6] This is neccesary as searching in the space of all possible weak classifiers of all 16 types proved to be too time consuming. Indeed, searching $M = 200$ times over one type of weak classifier space takes about 30min for one fold and one class on a 3GHz Pentium 4 PC.

Table 3. Mean results on boosting individual type of weak classifiers (first row) and their percentage of beeing choosed when combined with SVM

h_σ^k	h_σ^{kl}	$h_{\sigma=}^{kl}$	h_θ^k	h_θ^{kl}	$h_{\theta=}^{kl}$	$h_{\sigma\theta}^k$	$h_{\sigma\theta}^{kl}$	$h_{\sigma\theta=}^{kl}$	h_B^k	h_B^{kl}	$h_{k\cap l}$	$h_{k\in l}$	$h_{k\subset l}$	$h_{k\propto l}$	$h_{k\in\aleph_l^5}$	$h_{k\in\aleph_l^{10}}$
63.6	66.5	46.2	62.1	62.6	48.8	61.6	64.5	48.8	62.8	63.8	63.3	64.1	53.8	58.5	62.4	64.5
2	21.2	2.8	0.4	13.5	3.2	0.4	9	1.6	3.8	35.7	2.7	0.8	0.1	0.3	0.9	1.6

Fig. 3. Examples of the most relevant clusters (3) for h^k and the pair of clusters h_σ^{kl}, $h_{k\subset l}$ and $h_{k\cap l}$ respectively in case of chairs, road signs, boats and flowers. In case only green keypatches are shown means that we obtained $k = l$.

Table 4. Confusion matrix and mean ranks for SVM_{all}

true classes →	bikes	boats	books	cars	chairs	flowers	phones	r. signs	shoes	s. toys
bikes	**74.6**	1.6	1.5	0.6	5.5	1.2	0	1	0.6	0.8
boats	3.3	**81.8**	4.1	4.7	4.6	1.2	0.4	1	2.1	0.8
books	0	2.1	**78.2**	1.3	2.3	0	4.8	3.3	0.7	1.1
cars	3.8	3.7	2.2	**77.5**	7	0.4	2	3.3	1.3	0.4
chairs	10.4	2.8	4.4	4.8	**65.2**	2.1	2.8	4.3	2.5	0
flowers	1.2	0.9	0	0.6	1.8	**89.6**	0	1.4	0.6	0.8
phones	0.8	0.7	3	2.9	2.3	0.4	**76**	1	2.1	0.4
road signs	1.3	0.9	2.2	1.6	4.6	1.3	2.4	**76.2**	1.3	0
shoes	2.9	5.3	3.7	6	5.5	1.3	10.4	7.6	**83.8**	11.9
soft toys	0.4	0.2	0.7	0	1.2	2.5	1.2	0.9	5	**83.8**
mean ranks	1.4	1.3	1.4	1.3	1.8	1.2	1.9	1.5	1.5	1.1

Figure 3 illustrates examples of most relevant weak classifiers selected by boosting single type classifiers and Table 3 second raw the percentage of beeing choosed when combined with SVM through boosting. Mainly weak classifiers based on pairs of keypatches were selected.

We then combined the selected base learners across the different types through boosting. First the 17 type of geometry based weak learners were combined with hypotheses h_k and $h_{k,l}$. This (see fourth row of Table 2) slightly improved the boosting resuls without geometry (first two rows) but gave still much lower performance than applying the SVM. Therefore we rather combined the SVM outputs with geometrical

weak classifiers trough generalized AdaBoost (Table 2 fifth row). The SVM outputs were normalized to $[0, 1]$ using a sigmoid fit[7] [13] and then mapped to $[-1, 1]$.

The SVM performance was significantly improved. Table 4 shows the confusion matrix and the mean ranks (the mean positions of the correct labels when labels output by the multi-class classifier are sorted by the classifier score) for this combined classifier.

6 Conclusions

We have investigated how weak classifiers depending on geometric properties can be exploited for generic visual categorization. Results have been given on a challenging ten-class dataset which is publicly available. The benefits of the proposed method are its efficiency, invariance and good accuracy on a challenging dataset. Overall improvement in error rate has been demonstrated through the use of geometric information, relative to results obtained in the absence of geometric information.

While we have explored 19 types (17 with geometry) of weak classifier, many more can be envisaged for future work. Geometric properties are of course widely used in matching. It will be interesting to explore how recent progress in this domain such as techniques in [5,6] can be exploited for categorization. It will also be interesting to evaluate other approaches to boosting in the multiclass case such as the joint-boosting proposed in [16], which promise improved generalization performance and the need for fewer weak classifiers.

References

1. P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *Proc. ECCV*, volume 1, pages 350–362, 2004.
2. Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *JMLR*, 5:913–939, 2004.
3. G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.
4. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, volume 2, pages 264–271, 2003.
5. V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *Proc. ECCV*, volume 1, pages 40–54, 2004.
6. S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *Proc. BMVC*, volume 2, pages 959–968, 2004.
7. B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proc ECCV Workshop on Statistical Learning in Computer Vision*, pages 17–32, 2004.
8. Y. Li, J. A. Bilmes, and L. G. Shapiro. Object class recognition using images of abstract regions. In *Proc. ICPR*, volume 1, pages 40–44, 2004.

[7] This transformation of SVM outputs to confidence was also applied when we ranked the outputs from different classes

9. D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, 1999.
10. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*, volume 1, pages 128–142, 2002.
11. A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. ECCV*, volume 2, pages 71–84, 2004.
12. J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. GCap: Graph-based automatic image captioning. In *Proc. CVPR Workshop on Multimedia Data and Document Engineering*, 2004.
13. J. C. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.
14. G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for Adaboost. *ML*, 42(3):287–320, 2000.
15. R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
16. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *Proc. CVPR*, volume 2, pages 762–769, 2004.
17. V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

Content Based Image Retrieval Using a Metric in a Perceptual Colour Space

G.Tascini and A. Montesanto

Dip. Di Elettronica, Intelligenza Artificiale e Telecomunicazioniv,
Università Politecnica delle Marche
g.tascini@univpm.it

Abstract. The aim of the present work is building an evaluation method for the similarity between colour hues. The method is defined by studying the attribution process, by human subjects, of colour hue couple to similarity classes (from 'very similar' to 'little similar'). From the study of these categorical judgements it is derived that the relation between the hue and the colour similarity is 'not-isometric' and greatly depends on the colour category. This result allows to extract representative functions for the three colour of the subtractive system: Red, Yellow, Blue. Besides we used a new method for segmenting the colour, based on the similarity with the main colours. Our method defines a quaternary tree structure, named 'Similarity Quad-Tree'; it is capable of extracting, from the whole image, the belonging degree to the Red, Yellow and Blue colours and their similarity with the reference colour. The check on the method applicability has given good results both: in the user satisfaction and in the computation. The approach may be viewed as a simple and fast indexing method.

Keywords: Colour Perception, No isometric Similarity Metrics, Human Subjects, Content based image retrieval.

1 Introduction

Various fields**,** like art, medicine, entertainment, education, and multimedia in general, require fast and effective recovery methods of images. Among these are Content Based Image Retrieval, in which images are described not by keywords but by content**.** A main approach is using low level characteristics, like colour, for segmenting, indexing and recovering. This work presents a method for annotating and recovering images that uses a new evaluation method for the similarity between colour hues that corresponds to human colour perception. In addition a fast and effective method for image indexing, is presented. In literature many methods are presented to this aim. A simple and fast method is based on a set of key words that describes the pictorial content [1]. The drawbacks of this approach are various: the method is hard for the big data bases; the quality of key words is subjective; the search by similarity is impossible. A more general approach to multimedia recover, different from those based on visual or acoustic data. The main difference depends on extraction of features. A popular approach is the Query By Example (QBE), where

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 621 – 628, 2005.

the query is a key object, in particular an image, in the data base or depicted at query time. The content-based methods allow recovering images by the visual language characteristics, like similarity, approximation and metric relations, research key as figures, structures, shapes, lines and colours. As consequence they are many modes of indexing, storing, searching and recovering visual data. More refined are the methods in which the images may be analysed in the query phase; the corresponding software are called: Content Based Image Retrieval (CBIR) Systems. As the Query by Colour**,** two types of approaches are important: 1) retrieval of images with global colour distribution similar to the query image one, interesting for the pictorial data bases; 2) recovering of an object in a scene, using its chromatic features. [2] We will briefly describe some of most popular CBIR. QBIC, that means Query By Image Content [4], uses various perceptual characteristics and a partition-based approach to the colour. Introduces the Munsell transformation and defines a colour similarity metric [3]. The system is limited in the search of spatial characteristics. Virage [3] that supports the query about colour, structure and spatial relations operated on the following four primitives: Global Colour, Local Colour, Structure e Texture. Photobook [5] is an interactive set of tools developed at M.I.T. Media Laboratory on the Perceptual Computing. The system interacts with user by Motif interface. The matching is performed on the feature vectors extracted by considering invariance, scaling and rotation. VisualSEEk [6,7] and WebSEEk [8] are academic information systems developed at the Columbia University. VisualSEEk is a hybrid image recovery system that integrates the feature extraction using the colour representation, the structure and the spatial distribution. The recovering process is enhanced with algorithms based on binary trees. WebSEEk instead is a catalogue-based engine for the World Wide Web; it accepts queries on visual properties, like colour, layout correspondence and structure. ImageRover [9] is an image recovery tool developed at the Boston University. This system combines visual and textual queries for the computation of the image decompositions, associations and textual index. The visual features are stored in a vector, using colour and histograms texture-orientation; the textual one are captured by using the *Latent Semantic Indexing* on the association of the words contained in the HTML document 10]. The user refines the initial query using the *relevance feedback.* The Munsell colour space is a three-dimensional polar space, the dimensions being Hue, Value and Chroma. Value represents perceived luminance represented by a numerical coordinate with a lower boundary of zero and an upper boundary of ten. Chroma represents the strength of the colour, the lower boundary of zero indicating an entirely achromatic colour such as black, white or gray. The upper value of Chroma varies depending upon the Value and Hue coordinates. The Hue dimension is polar and consists of ten sections that are represented textually each with ten subsections represented numerically, as shown in Fig. 1. Our work considers the only dimension Hue, while maintains constant the other 2 variable Saturation and Intensity. Differently from the Munsell colour space, it considers the space of this single dimension ‘not-uniform’ and ‘not-linear’, that is ‘not-idometric’. A main difference with the Munsell space is the following: our work do not evaluates the belonging of a colour to a ‘nominal’ category, that may be invalidated also by conceptual structures related to social history of examined population. We evaluate how mutch it is similar to a target-colour a variation of it, performed in the only tolnality dimension. Then the not-linearity is related to the colour similarity and not

their categorization. The results are related to the subject judgements on the similarity evaluation between two colours, and not on the hue. If the Munsell space is perceptually uniform and linear, then a variation Δh of hue would be proportional to the related similarity variation Δs : the results have shown that this direct proportionality, between hue-variation and similarity-variation, of two colours do not exists.

2 The Similarity Function

A test on humans has been performed to the aim of defining a colour-similarity function. To appraise the similarity with the image target they are presented, to the human subjects, 21 images that are similar but perturbed in the colour. Such images maintain spatial fixed disposition. Everything is repeated for the three colours used in the search: *yellow*, *red* and *blue*. They are the three fundamental colours of the subtractive model..

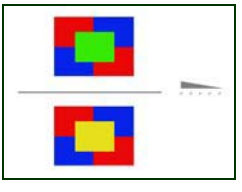


Fig. 1. The test image for the perception of the colours

2.1 Human Evaluation of the Similarity

The experiment on colour similarity evaluation uses the images shown in the Fig.1. They vary only the colour of the central rectangle. An image target and its variation are shown simultaneously to the subject. 63 couples of images are randomized for each proof (each set of 21 corresponds to a target image). The subject sees all the mixed images that are not subdivided for typology of basic colour. The subject have to assign a coefficient of "similarity to the target" of the images using scale from 1 to 5: 1= nothing, 2= very little, 3= little, 4= enough and 5= totally. Once submitted the test of similarity evaluation of the 12 subjects, we compute the assign frequencies of the 63 stimuli to the 5 categories. The average answer distributions, for each human subject, are not homogeneous and very asymmetric. The trend of the frequencies of the judgments of similarity for each colour furnishes an idea of how the different colours are represented at the cognitive level. We apply to the frequencies a unidimentional scaling to have a precise measure of the distance between the variation steps for each colour. The choice of the single dimension is because we consider only the variation of the hue, while the others two dimensions are fixed. The scaling is applied singularly to each scale of colour, so we have a scale of measurement for the Red, one for the yellow and one for the Blue. In this job the hypothesis is that the colours are different between them: 1- for the physical property of the bright energy, different for each one, 2- each colour has perceived psychologically in different

manner and 3- the same stimuli are different entity. The colours could not be measured on the same scales, but they are subject to some transformations considering the verse that point out the affiliation to a particular category. The unidimentional scaling has been able to quantify the qualitative data of the first level: comparing the graphic of the two levels underlines their homogeneity. We have to notice that the graphic of the first level are not identical to those of the second level in as these last is more precise for what concern the metrics. Now we need a model for representing the relationship between hues (Hue with S=constant, I=constant) and related values of similarity. A not linear regression is used for finding a function (y) able to interpolate the points of the similarities related to the hues (x). This function is a polynomial. The values of the single parameters are different according to the derivation; each colour will weigh the value of similarity of the variation of tonality in different manner for the different values of the parameters. The resultant functions are the followings:

$$\text{Yellow: } y = -.00026 * x^3 + .046504 * x^2 - 2.6155 * x + 43.9572; \quad (1.1)$$

$$\text{Blue: } y = .000061 * x^3 - .02622 * x^2 + 3.64571 * x - 163.38; \quad (2.2)$$

$$\text{Red: } y = .000001 * x^3 - .00096 * x^2 + .198553 * x - 4.2372. \quad (2.3)$$

We underline the discrepancies between the form of the functions and the real disposition of the points in the space; to overcome this problem we have defined some broken functions to have a better representation of the data. So we have different range of tonality (h= hue) and they are divided for the three fundamental select colours: Blue (100< h< 204),(**2.1**); Yellow (26< h< 78),(**3.2**) and Red (-42< h< 22),that coincides to (0< h< 22) OR (198< h< 240),(**3.3**). Using these functions, given a value of hue with saturation of 240 and brightness of 120, we can see to which colour this hue is similar and how much is similar to the reference colour. For example: h= 0==> Red with sim.=- 3.199093,(**4.1**); 0<h<26==>Red with sim.= 0.016065* h2- 0.16173* h- 3.1260, (**4.2**) and 25<h<42==> Yellow with sim.=- 0.03198* h2+ 1.83212* h- 25.677, (**4.3**).

3 Similarity Quadtree

Once defined the relationship between Hue and similarity degree with reference colour, it is possible to represent an image using these concepts. To this purpose we have defined a new data-structure, named 'Similarity Quad-tree'. In practice we depart from the original image and we recursively partition it quadrants: the procedure continues until a homogeneous quadrant is reached. We can stop the process also at a given level of subdivision: in this case the quadrant is not uniform: we can assign to the quadrant a colour equal to: a) prevailing colour, b) average colour. After this we have a value of Hue for each quadrant. At this time we calculate, for each quadrant, the value of similarity of the given Hue with the reference colour. The Similarity Quad-Tree (SQT) so obtained constitutes an image representation, related to the subjectivity with which the human perceives the similarity between colours; SQT have shown its usefulness in retrieval operations. In our colour based recovery we use two steps: 1) a decomposition of the image in a quad-tree structure, 2) matching

between the obtained data-structure and a reference one. The whole procedure includes a series of operations; among these are size adjusting and pixel grouping; after we build a quaternary tree representing in each leaf the content in terms of primary colours Red Green Blue, named *RGB Quad Tree*: each leaf contains a string of three values, Red Green Blue. Then the *Similarity Quad- Tree* is built, departing from the RGB values, in which a leaf do not represent the colour but the values of the *similarity of the Hue* respect to the reference colour of the quadrant. The other nodes simply contain pointers to the children using the cardinal points notations: North West (NO), North East (NE), South East (SE) and South West (SO). The three colours of the SQT are defined in the RYB (Red, Yellow, Blue) space, while the *reference values* are extrapolated with the aid of our scaling computation on similarity judgments frequencies: a) *Red: ==> ref. =- 3,199093; b) Yellow: ==> ref. =- 3.544842;* c) *Blue: ==> ref. =- 4.768372.* The procedure for computing the similarity receives a leaf-node from an RGB_QuadTree in input extracts the three RGB values and from these computes the value of the hue (h). After the Hue range indicates the reference colour (3.1), (3.2), (3.3), as previous seen. Then the similarity to the main colour is computed, using the functions (2.1), (2.2) and (2.3) previous defined. The main colours are represented with the values 0, 1, 2 that stand respectively for *red*, *yellow* and *blue*. If the extracted colour from the RGB_QuadTree is perceptually not valid (for instance: little saturated or too bright), it is not inserted in the leaf. We have computed the hue of a given RGB colour, adopting a constant saturation value of 240 and a brightness of 120, corresponding to 1 and 0.5 in relative coordinates. The main procedure, that concerns the creation of the similarity tree, receives in input the roots of both trees: RGB_QuadTree and Similarity_QuadTree. When the RGB_QuadTree has reached a leaf-node, then the procedure insert the value of similarity in the Similarity_QuadTree. When it is not at a leaf-node the four children of the node are created. The SQT is suitable for indexing: here we associate with the database of imagines a file containing the data derived from the segmentation. This file is created before the retrieval, in such a way allowing fast access when these features are required. The structures of data-file will be composed by a sequence of information organized as in figure 5. An image may be selected for being used as query; with this the figure properties, of all database, will be compared to obtain a similarity matching. Our software gives the opportunity of sketching an image through a Java application. The user can also insert an image or a photo as query. The Matching process computes a score that identifies the similarity of a generic image with a selected one like key of search, assigning a value from 0 (completely different image), to 10000 (perfectly coincident image). This last value is divided in the following way: a) 4000 points derive from an appropriate comparison with the order of quantity of the principal red, yellow and blue colours; b) 6000 points result from the difference between the similarities of the two images for every principal colour. If the first two colours have identical quantity, then we assign to the variable score 2000 credits, while if also the seconds are equal then we assign 1250 credits, and so on. This to overcome the inconvenience of judging very similar two images only because they is a pixel of a certain colour similar to the average of all the colours of the other.

4 Results

We use two kind of image like keys of search in a small data-base. A subject gives a perceptive judgment on the similarity of the images to the keys, by concluding retrieval: this may be considered the optimal result that our program should reach. Then we perform the same two searches using our software. Fig. 2 show the images choices as keys of search. The subjects assign a score from 1 to 10 for each element of the database, where 10 stands for the best similarity and 1 for the smallest one. The total score associates a meaning degree of similarity to all database respect to the key image. Figure 3 shows the data-base images used in the example.

Fig. 2. Image key 1 and Image key 2

Fig. 3. The 10 images of the example

4.1 Comparison Between Human Decision and Software

In the tab 1 it is listed the numbers of the sorted figures, departing from the more similar to the key image 1 up to the less similar. The score that we find in the last column it is the associated value to the position of the orderly images from our CBIR system as regards the order of the same result from the Test. The used rule is that of assign 10 points to the images that they are in the same row, while two points for each line of distance are scaled. The last row corresponds simply to the percentage calculus of the score, therefore represents the empirical evaluation of the percentage of

correctness in the measure of similarity of the software. This technique allows to have a range of percentages from 0 to 100 points because allows of assign also score negative, therefore the value gotten like a valid respect of the correctness of the created program could be considered. In tab.1 we compared the two methodologies using first as key the photo of the test 1. In this case we obtain a score of 88%, which points out that the program possesses a good analogy with the real perception human. Even in tab.1 we can see the test numbers 2, where the image key is a sketch. We used the same methods for the calculus of the score. The percentage of correctness in the measure of similarity of the software is 80%, so we have a good level of fidelity.

Table 1. Results of testing the two key images

Image key 1					**Image key 2**			
Similarity Order	***Image n°***		**Score**	**Similarity Order**		***Image n°***	**Score**
	HUMAN	**SOFTWARE**			**HUMAN**	**SOFTWARE**	
1	5	5	10	***1***	10	10	10
2	3	3	10	***2***	2	2	10
3	2	2	10	***3***	5	3	8
4	1	10	8	***4***	3	5	8
5	10	1	8	***5***	1	9	8
6	8	8	10	***6***	8	1	6
7	9	6	8	***7***	9	6	6
8	6	9	8	***8***	7	8	8
9	7	4	8	***9***	6	7	6
10	4	7	8	***10***	4	4	10
Percentage of fidelity:			**88%**	***Percentage of fidelity:***			**80%**

5 Conclusions

The paper has presented a method of Content Based Image Retrieval, whose originality is related to two main aspects: 1) the definition of a *perceptual approach* that allows to build a new method for the similarity between colour hues evaluation, and that represents the abstraction of the content in a simple and efficient way; 2) the introduction of a new *methodology for indexing* the images, based on the Similarity Quad-Tree. So we can extract the properties only related to the colour, excluding the features like form and spatial relationships between objects of the image. The efficiency of this idea derives from the similarity function application. This function is derived from experimental evaluation of the perceptual metrics used by the human while judge the similarity between colours. The indexing methodology quality is related to a fast access to the representative features of an image that are stored in a vector: this necessarily involves high computation speed and cost minimization. Preliminary results give on 8000 image data-base, about 15 seconds of image-seek, for an ancient Pentium III at 750 MHz, where about 9 seconds are used for loading

graphic interface, 2 second for feature extraction and less the 4 seconds for searching in the data-file and for matching. In a commercial PC with over 3 GHz of clock we go down under 2 second for all computation.

References

1. Y. Rui, T. S. Huang, S. F. Chang – "Image retrieval: Past, present, and future" - *Journal of Visual Communication and Image Representation* – pages 1÷23 - 1999.
2. J. R. Smith – "Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression" - *PhD thesis, Graduate School of Arts and Sciences, Columbia University* - 1997.
3. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, C. F. Shu – "The Virage image search engine: An open framework for image management" - *In Proceedings of the Storage and Retrieval for Still Image and Video Databases IV* - pages 76÷87 - San Jose, CA, USA - February 1996.
4. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker – "Query by image content and video content: The QBIC System" - *In IEEE Computer* - pages 23÷32 - September 1995.
5. Pentland, R. Picard, S. Sclaroff – "Photobook: Content-based manipulation of image databases" - *In International Journal of Computer Vision* - pages 233÷254 - San Jose, CA, USA - June 1996.
6. J. R. Smith, S. F. Chang - *Querying by Color Regions using the VisualSEEk Content-Based Visual Query System* - Intelligent Multimedia Information Retrieval, AAAI/MIT Press - 1996.
7. J. R. Smith, S. F. Chang – "VisualSEEK: a fully automated content-based image query system" - *In Proceedings of the 4th ACM International Conference on Multimedia* - pages 87÷98 - Boston, Massachusetts, USA - November 1996.
8. J. R. Smith, S. F. Chang – "Visually searching the web for content" - *IEEE Multimedia Magazine* – pages 12÷20 - 1997.
9. S. Sclaroff, L. Taycher, M. La Cascia – "Imagerover: A content-based image browser for the world wide web" - *In Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries* - pages 2÷9 - San Juan, Porto Rico - June 1997.
10. M. La Cascia, S. Sethi, S. Sclaroff – "Combining textual and visual cues for content-based image retrieval on the world wide web" - *In Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries* - pages 24÷28 - Santa Barbara, CA, USA - June 1998.

Efficient Shape Matching Using Weighted Edge Potential Functions

Minh-Son Dao, Francesco G.B. DeNatale, and Andrea Massa

DIT - University of Trento,
Via Sommarive, 14 - 38050 Trento, Italy
{dao, denatale, massa}@dit.unitn.it

Abstract. An efficient approach to shape matching in digital images is presented. The method, called Weighted Edge Potential Function, is a significant improvement of the EPF similarity measure, which models the image edges as charged elements in order to generate a field of attraction over similarly shaped objects. Experimental results and comparisons demonstrate that WEPF enhances the properties of EPF and outperforms traditional similarity metrics in shape matching applications, in particular in the presence of noise and clutter.

1 Introduction

The task of automatically matching shapes in digital images is a fundamental problem in pattern recognition. Applications of shape matching include industrial processes, robotics, video surveillance, and many others. Several approaches have been proposed in the literature to solve this problem in different application domains. For a thorough survey on the matter, please refer to [7]. The above mentioned applications are concerned with two different problems: (i) how to match objects, and (ii) how to measure the similarity among them. The first focuses on the matching procedure between an object and a model, while the latter concentrates on the problem of defining when a target is ẗreasonablyšimilar to a query object. Although the two aspects are often strongly connected, the definition of effective similarity measures is gaining increasing attention, also due to emerging applications such as content-based image indexing and retrieval.

As far as the evaluation of the similarity is concerned, several metrics have been developed. Traditional metrics include Minskowski, Euclidean, and Mahalanobis. More recently, similarity measures based on fuzzy logic have also been proposed. For a thorough survey on the matter, please refer to [8]. An important class of methods is the one based on distance transforms. Chamfer Matching [2] and Hausdorff Distance [5] are the current reference approaches in the field, and ensure very good performance even in the presence of complex images, distortion (e.g., affine transforms), occlusion, noise, and clutter. In [3], an alternative approach was proposed called Edge Potential Function (EPF). This method mimics the attraction field generated by charged particles [6], and differs from traditional point-to-set distances in the fact that it exploits the all edge instead

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 629–636, 2005.

of just nearest neighbors. This allows reinforcing the effect of coherent contours as compared to noise.

This paper reports a significant improvement of EPF, the Weighted Edge Potential Function (WEPF), which shows interesting properties for efficient shape matching. The paper is organized as follow: in Sect. 2 the concepts of EP and WEP are outlined and motivated. In Sect. 3 the use of WEP to shape matching under distortion conditions is presented. In Sect. 4, a set of selected test results is proposed, showing the performance of the proposed approach in different application conditions, and comparing it to other established approaches.

2 Weighted Edge Potential Functions

The concept of EPF can be summarized as follows (see [3] for details):

Definition 1: Given a test point q and a set of edge points $A = \{a_1, \ldots, a_m\}$, the edge potential EP generated in q by A is:

$$EP(q, A) = \begin{cases} \frac{1}{\epsilon} \sum_{i=1}^{m} \frac{1}{\|q - a_i\|}, \forall a_i : a_i \neq q \\ \gamma, \exists a_k : a_k = q \end{cases} \quad (1)$$

where γ is a peak value replacing the singularity point of the potential, and is a permittivity constant that controls the slope of the potential.

Definition 2: Given two finite point sets $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ the EP Function (EPF) of set B with respect to set A is defined as:

$$EPF(B, A) = \frac{1}{|B|} \sum_{b_i \in B} EP(b_i, A) \quad (2)$$

Given $\gamma = 1$, EPF(B,A) is in the range [0:1] where 0 reflects complete similarity between B and A, while 1 indicates a perfect match, thus providing a normalized similarity measure. It is important noting that EPF is asymmetric. This means that if for instance B is a subset of A, we find a perfect matching of B over A (EPF=1), but not of A over B. This allows matching correctly fragments or subparts of shapes.

Although EPF was demonstrated to be a very efficient similarity measure, it can be improved in two respects:

- EP depends on the values γ and ϵ, which are heuristically chosen.
- Though the high slope helps the convergence to be faster and isolates noise spots, it makes the EPF fall suddenly when the set A misses some points due to occlusion, clutter or deformation.

The second problem can be particularly critical in the presence of inaccurate preprocessing and edge-extraction, where several contour pixels may get lost. In

order to overcome these problems, a weighted version of the EP is proposed, by introducing an adaptation (weighting) parameter:

Definition 3: Given a test point q and a set of edge points $A = \{a_1, \ldots, a_m\}$, the weighted edge potential WEP generated in q by A is:

$$WEP(q, A) = w_q EP(q, A) = w_q \left(\frac{1}{\epsilon} \sum_{i=1}^{m} \frac{1}{\|q - a_i\|} \right) \tag{3}$$

where $w_q = \epsilon min_{a_i \in A} \|q - a_i\|$ is the weighting factor. It is easy to verify that WEP is independent of ϵ. To this purpose, let's define $r_{qm} = \|q - a_m\| = min_{a_i \in A} \|q - a_i\|$

After a few simple passages we obtain:

$$WEP(q, A) = 1 + r_{qm} \sum_{i=1}^{m-1} \frac{1}{\|q - a_i\|} \tag{4}$$

If WEP is computed at every image point, a surface is generated with zero height in correspondence of each edge point. It is to be observed that when q is far from its nearest neighbor, the weighting factor becomes negligible. In order to better compare WEP with EP, WEP is normalized and remapped, according to Eq. 5

$$NWEP(q, A) = f(WEP(q, A) - 1) \tag{5}$$

Theorem: Given a finite point set $A = \{a_1, \ldots, a_m\}$ and a point q, if we add to A a finite point set B made up of k points to create a finite point set $C = A \cup B$, then $WEP(q, C) \geq WEP(q, A)$.

Proof: Let's consider $r_{qm} = \|q - a_m\| = min_{a_i \in A} \|q - a_i\| \leq min_{b_i \in B} \|q - bi\|$ if $r_{qm} = 0$ then the theorem is evidenced. If $r_{qm} \neq 0$, then we have:

$$WEP(q, C) = min_{c_i \in C} \|q - c_i\| \sum_{i=1}^{m+k-1} \frac{1}{\|q - c_i\|} = r_{qm} \sum_{i=1}^{m+k-1} \frac{1}{\|q - c_i\|}$$

$$= r_{qm} \left(\sum_{i=1}^{m-1} \frac{1}{\|q - a_i\|} \right) + r_{qm} \left(\sum_{j=1}^{k} \frac{1}{\|q - b_j\|} \right)$$

$$WEP(q, C) = WEP(q, A) + WEP\left(q, B^{'}\right), B^{'} = B \cap \{a_m\}$$

Thus, the theorem is evidenced. In the case $min_{a_i \in A} \|q - a_i\| \geq min_{b_i \in B} \|q - b_i\|$, A and B are permuted, and the theorem is evidenced as well.

This theorem introduces an important advantage of the weighted function, consisting in a higher robustness to noise and clutter. As a matter of facts, in the presence of dot noise the weighting factor produces an automatic increase of the slope of WEP surface, which tends to isolate noise spots. On the contrary, in the presence of clutter or scattered contours, WEP automatically decreases

the slope, thus producing a reasonably continuous potential function even along discontinuities in the contour.

Definition 4: Given two finite point sets $A=\{a_1,\ldots,a_m\}$ and $B=\{b_1,\ldots,b_n\}$, the Weighted EPF (WEPF) and the Normalized WEPF (NWEP) of B over A are defined as:

$$WEPF\,(B,A)=\frac{1}{n}\sum_{i=1}^{n}WEP\,(b_i,A) \tag{6}$$

or $NWEPF\,(B,A)=\frac{1}{n}\sum_{i=1}^{n}NWEP\,(b_i,A)$. Also in this case, $NWEPF\,(B,A) \neq NWEPF\,(A,B)$, and the larger the NWEPF value the higher the similarity of set B with respect to set A.

3 WEPF and Shape Matching

Several interesting pattern recognition problems can be addressed by using the proposed methodology. Here, we will consider the problem in its more general form, while specific applications have been already proposed in previous works with application to image retrieval[3] and video indexing[4]. The objective is to determine if the target image contains an object whose shape is similar to a model after an affine transform. The affine transform produces an instance of the model by taking into account an operator $c=(t_x,t_y,\theta,t_w,t_h)$ where θ is rotation; t_x, and t_y are the translation along x-axis and y-axis, respectively; and t_w, and t_h are the scaling along x-axis and y-axis, respectively. The matching process consists in determining the operator c that maximizes the similarity of the relevant instance of the model and the target. As far as the definition of a suitable metric is concerned, the matching function defined in Eq. 6 can be rewritten as:

$$WEPF\,(c_k)=\frac{1}{n^{(c_k)}}\sum_{i=1}^{n^{(c_k)}}WEP\left(b_i^{(c_k)},A\right) \tag{7}$$

or $NWEPF\,(c_k)=\frac{1}{n^{(c_k)}}\sum_{i=1}^{n^{(c_k)}}NWEP\left(b_i^{(c_k)},A\right)$, where $n^{(c_k)}$ is the number of pixels of the c_k-th instance of the sketch contour, $b_i^{(c_k)}$ is the i^{th} pixel of the c_k instance.

Eq. 7 can be considered a highly nonlinear multivariate fitness function to be globally maximized. This process can be solved in different ways by taking into account convergence and speed criteria. Multi-resolution and hierarchical approaches should be used to this purpose, as well as statistical methods (e.g., simulated annealing, genetic algorithms). In the present work, a Genetic Algorithm (GA) was implemented and customized to this purpose, providing a very efficient optimization tool in terms of speed and reliability.

4 Experimental Results and Conclusions

In this section, the results achieved by WEPF are analyzed and compared with other established methods. First, a comparison is proposed in terms of performance of the similarity measure (i.e., capability to correctly catch the similarity/differences among two shapes in complex/noisy scenarios). Then, a possible application of WEPF to sketch-based image matching is assessed.

As far as the comparative analysis is concerned, two well known approaches are used as a reference: Chamfer Matching and Hausdorff distance. To this purpose, the following schemes are taken into consideration:

- (DT, CM): Chamfer Matching using Distance transform [1]
- (DT, HD): Hausdorff using Distance transform.
- (EPF): Edge Potential Function.
- (NWEPF): Normalized Weighted Edge Potential function.

Two test cases are investigated to show the beneficial attributes of EPF: in both cases a complex scenario is considered, with various shapes mixed to different color textures. In the first test case, images are processed under severe noise conditions. In the second one, edges are artificially damaged.

Test 1: Fig. 1 shows a sketch-based shape matching: images (a) and (b) are the query and the target images, while (d) is a noisy version of the target image. (c) and (e) show the edge maps extracted from (b) and (d) by a Canny-Rothwell detector.

To demonstrate the effectiveness of the proposed similarity measure, an exhaustive search was performed by making the transformation operator to vary in a large range. The goal was to verify if the peak of the similarity function corresponds to the optimum matching, and if there are suboptimal or wrong solutions achieving near fitness.

Fig. 2 shows the histogram of the similarity values achieved for each examined position: it should be noticed that distance-transform-based approaches generate a spike in the histogram near the maximum similarity values, thus meaning that numerous solutions assume values near to the maximum. This makes the

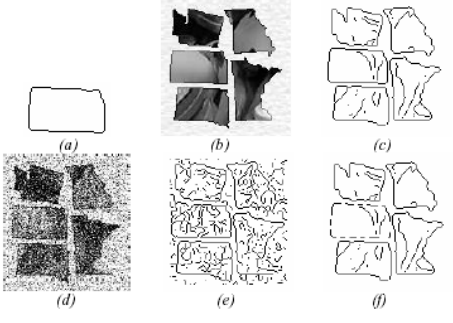


Fig. 1. Testbed 1

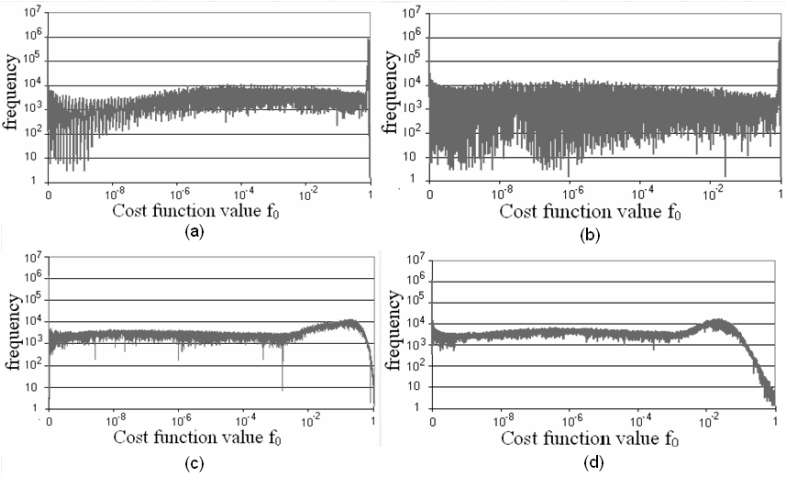


Fig. 2. Exhaustive search results - noisy environment (a) f(x)=(DT,CM) (b) f(x) = (DT, HD) (c) f(x) = (EPF) (d) f(x) = NWEPF. Frequency = Number of trial solutions for which f(x) = f_0

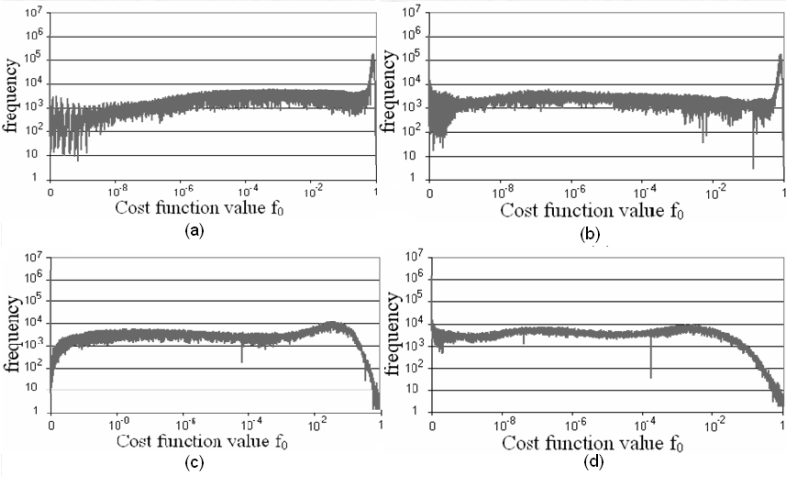


Fig. 3. Exhaustive search results - clutter environment (a) f(x)=(DT,CM) (b) f(x) = (DT, HD) (c) f(x) = (EPF) (d) f(x) = NWEPF. Frequency = Number of trial solutions for which f(x) = f_0

convergence more difficult, as confirmed by the first part of Fig. 4, where the number of false positives (wrong positions showing a computed fitness higher than the correct solution) is analysed. In the example only NWEPF is able to detect the best match with no false positives.

Test 2: Let us consider again the case of Fig. 1. This time, we would like to compare NWEPF with competing algorithms in the case of imperfect edge

Test Example	Method	The largest value of similarity measure	Frequency	Position status
Ex. 1 Noisy environment	(DT, CM)	0.998590	3	1 right, 2 wrong
	(DT, HD)	0.998588	3	1 right, 2 wrong
	(EP, EPF)	0.980825	2	1 right, 1 wrong
	(NWEP, NWEPF)	0.916814	1	1 right
Ex. 2 Clutter environment	(DT, CM)	0.997416	4	1 right, 3 wrong
	(DT, HD)	0.997407	4	1 right, 3 wrong
	(EP, EPF)	0.933193	1	1 right
	(NWEP, NWEPF)	0.914820	1	1 right

Fig. 4. Comparing the efficiency and effectiveness among (DT, CM), (DT, HD), (EPF) and (NWEPF) methods

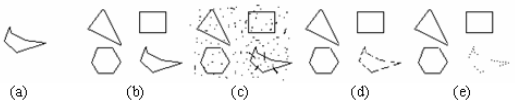


Fig. 5. Testbed 2

extraction. To this purpose, we simulate the loss of contour points by randomly erasing 50% of the contour points on the target object, i.e., the object that matches the query (f). Also in this case an exhaustive search is performed.

The resulting histogram of similarity values (Fig. 3) demonstrates the higher selectivity of EP-based measures, and the relevant matching results proposed in the second part of Fig. 4 confirm once again the ability of EPF and particularly NWEPF to achieve almost perfect detection.

Sketch-based image matching: As an additional performance test, WEPF has been introduced as a similarity measure in a content-based image retrieval (CBIR) scheme based on a genetic matching [3]. The comparative performance analysis was carried out by substituting different similarity metrics in the same matching scheme.

In particular, the following scenario is investigated: detection of the presence of a user-given sketch within a binary image representing some object shapes with added noise and clutter. Moreover, comparisons with state of the art approaches are provided to show the effectiveness of the proposed approach. To demonstrate the robustness of EPF with respect to the matching strategy adopted and to the parameter setting, all the tests shown in this section are performed by using the same matching procedure, based on a Genetic Algorithm optimization, and a fixed set of parameters.

The scenario considered concerns the detection of a object shape under several noise conditions such as additive random noise and contour losses (with loss ratio ranging from 20% to 70%). Fig. 5 shows a typical example: Fig. 5a is the query model, which is applied to the target image in Fig. 5b. Fig. 5c shows the relevant noisy image, while Figs. 5d-e show the result of a 20% and 70% loss, respectively.

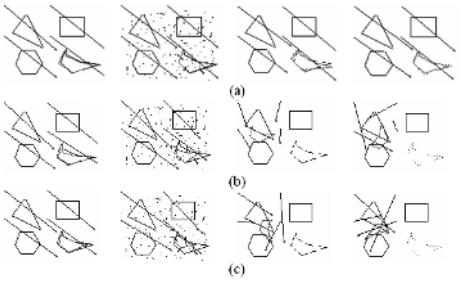


Fig. 6. (a) using (NWEPF) (b) using (DT,CM) (c) using (DT, HD)

Figs. 6a, b, and c illustrate the result when using NWEPF, DT-CM, and DT-HD, respectively, to perform the matching. By analysing these figures, it is possible to clearly state that NWEPF achieves better performance in all the situations. In particular, the charts that show the GA performance make evident that the probability of falling in a local minima corresponding to a wrong object location is pretty high when using DT-HD and DT-CM, thus achieving a wrong positioning even in the presence of a high fitness value.

References

1. Borgefors, G., 1984. Distance transformations in arbitrary dimensions. In Com-puter Vision, Graphics, and Image Processing, vol.27, pp. 321-345.
2. Borgefors, G., 1988. Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm. In IEEE Transactions on Pattern Analysis and Matching Intelligence, vol. 10, no. 6, pp. 849-865.
3. Dao, M.S., De Natale, F.G.B., Massa, A., 2003. Edge potential functions and genetic algorithms for shape-based image retrieval. In Proceedings of IEEE In-ternational conference on image processing (ICIP'03), vol. 3, pp. 729-732.
4. Dao, M.S., De Natale, F.G.B., Massa, A., 2004. MPEG-4 Video Retrieval using Video-Objects and Edge Potential Functions. In Lecture notes of Pacific-Rim Conference on Multimedia.
5. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J., 1993. Com-paring Images Using the Hausdorff Distance. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no.9, pp. 850-863.
6. Stratton, J.A., 1941. Electromagnetic Theory. McGraw-Hill Book, NY.
7. Veltkamp, R.C., Hagedoorn, M., 2000. State-of-the-Art in Shape Matching. In Principles of visual information retrieval, Springer-Verlag, London, UK, ISBN:1-85233-381-2, pp. 87-119.
8. Van der Weken, D., Nachtegael, M., Kerre, E.E., 2002. An overview of similar-ity measures for images. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02), 13-17 May 2002, vol. 4, pp. IV-3317 - IV-3320.

Soccer Videos Highlight Prediction and Annotation in Real Time

M. Bertini, A. Del Bimbo, and W. Nunziati

Università di Firenze, 50139 Firenze, Italia
bertini, delbimbo, nunziati@dsi.unifi.it

Abstract. In this paper, we present an automatic system that is able to forecast the appearance of a soccer highlight, and annotate it, based on MPEG features; processing is performed in strict real time. A probabilistic framework based on Bayes networks is used to detect the most significant soccer highlights. Predictions are validated by different Bayes networks, to check the outcome of forecasts.

1 Introduction

Sports videos are particularly important because of the extremely large audience: broadcasters produce huge amount of video that cover sport events every day. Broadcasters need to select the most relevant sequences from sports video documents ("highlights") for different purposes:

- Archiving for later reuse (posterity logging)
- Production of programmes in short time (live logging)
- Selective transmission of events to handheld devices and smart phones (real time semantic transcoding)

The latter purpose is becoming more and more interesting since this service is now provided by several mobile phone companies. In order to be able to provide an effective selection of the most interesting events of a video there is need of an automatic annotation system, that should perform real time analysis of an incoming video stream, marking the beginning of a possibly interesting sequence (containing an highlight) and then signaling the end of the interesting sequence. Possibly the system should be capable to even forecast a highlight (e.g. to provide real-time services such as mobile phone access) that will last some seconds with a certain probability. The probability may be used by the final user in order to select only certain highlights that are forecast with a minimum probability.

Since we are addressing a system that forecasts an highlight all the processing has to be performed in real-time. To this end we have considered a set of cues and a system architecture that allows to perform RT processing. A method to extract rapidly visual cues is to use features extracted from the compressed domain, e.g. MPEG motion vectors and MPEG DC components of DCT blocks.

In Sect. 2 we report on previous work done in the field of sport video annotation. Discussion of the usage of Bayesian Networks for our particular task is provided in Sect. 3. The description of our proposed approach is provided in Sect. 4. Results are discussed in Sect. 5, and conclusions in Sect. 6.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 637–644, 2005.

2 Previous Work

Automatic sports video annotation has been addressed by several authors, with increasing attention in the very recent years. In particular rule-based modelling of complex plays for basketball is presented in [1] and in [2]. In this latter paper, basketball game shots are classified into one of three categories and basket highlights are detected from this classification, observing the occurrence of appropriate sequences of instances of these classes. In [3] Bayes networks have been used to model and classify American football plays using trajectories of players and ball. However, trajectories are entered manually, and not automatically extracted from the video stream. Kijak et al. [4] have used multimodal features to analyze tennis sports video structure. Models are used to integrate audio and visual features and perform stochastic modelling. Visual cues are used to identify the court views. Ball hits, silence, applause and speech help to identify specific events like scores, reserves, new serves, aces, serves and returns. Annotation of soccer videos has been addressed by a large number of researchers. Choi et al. [5] detect and track the ball and the players in the video sequence. The metric position of the players on the playfield is estimated using an approximation of the perspective planar transformation from the image points to the playfield model. In [6], the playfield is divided into several distinct zones. The framed zone is identified using patterns of the playfield lines which appear in the image. The ball position is also used to perform detection of shot on goal and corner kick events. In [7] MPEG motion vectors are used to detect events. In particular, they exploit the fact that fast imaged camera motion is observed in correspondence of typical soccer events, such as shot on goal or free kick. Recognition of relevant soccer highlights (free kicks, corner kicks, and penalty kicks) has been presented in [8]. Low level features like the playfield shape, camera motion and players' position are extracted and Hidden Markov Models are used to discriminate between the three highlights. More recently, in [9], Ekin et al. have performed event detection in soccer video using both shot sequence analysis and visual cues. In particular, they assume that the presence of highlights can be inferred from the occurrence of one or several slow motion shots and from the presence of shots where the referee and/or the goal post is framed. In [10] a system based on FSMs, that detects several different soccer highlights such as shot on goal, placed kicks, forward launches and turnovers, using visual cues has been presented. Ball trajectory is used by Yu et al. [11]. In order to detect the basic actions and compute ball possession by each team. Kalman filter is used to check whether a detected trajectory can be recognized as a ball trajectory. Experiments report detection of basic actions like touching and passing. Examples of detection of basic highlights in volleyball, tennis and soccer are reported. [12] has reported on detection of Formula 1 highlights using a multimodal fusion of cues and dynamic Bayes networks.

3 Probabilistic Highlight Modeling

Soccer highlights have a loosely defined structure. To capture the high intra–class variation that characterize the visual appearance of these events, we modeled

highlights using Bayesian networks (BNs). Bayesian networks [13] are directed acyclic graphs whose nodes represent random variables and whose edges correspond to direct dependencies between the variables. These dependencies are represented in a quantitative manner through conditional probability distributions. Among the reasons that make BNs appealing for our problem, the following are the most important:

- Factorization of the joint probability model. BNs represent the joint probability distribution defined by all possible points in the feature space into local, conditional distributions for each variable given its parents.
- Reasoning under missing observation. A BN is always able to produce an output, using all the evidence available. It does not require explicitly that all the observations are available. Moreover, even if observations are non–synchronized, the network still produce a valid output, hence different pieces of evidence can be gathered over time.
- Probabilistic output. The output of a BN is usually the posterior probability of an unobserved node, given the observations. This output can be directly related to user–centered preferences and needs.

For our particular task, a remarkable additional advantages of using BNs, stems from the causal interpretation that is usually associated to an edge in a BN. This give us a method to translate our knowledge into valid models. A top–down approach is adopted, which correspond to see observable features as directly "generating" higher level semantic events. We begin by defining a random variable for each of our observed feature, and a boolean random variable for the "highlight" node, which will tell us eventually whether an highlight is occurring or not. Directly connecting feature (input) variables to the output would result in a single conditional cpt, that would require us to specify a large number of parameters. To further factorize the joint probability distribution, we introduce additional, intermediate–level variables on which observed feature have a direct impact. To keep inference computation tractable for exact inference algorithms, we avoid to introduce cycles in the underlying undirected graph, ending in the reversed tree–like structure of fig.3 and 4.

Model parameters (i.e., probabilities in the CPTs) have been learned from labeled examples in a supervised way, as follows. Given N the number of a labeled example and $n(x)$ the number of time we observed event x, we use the following estimates for prior and conditional probabilities respectively:

$$P(x) \leftarrow n(x)/N \quad P(x|y) = P(x,y)/p(y) \leftarrow n(x,y)/n(y)$$

4 Real-Time Annotation

MPEG videos are used in order to extract as much visual features as possible from the compressed domain, to speed up the processing. In particular the system has been tested using MPEG-1 and MPEG-2 videos. Output of the BNs is used to detect interesting highlights, associating a confidence number to the beginning

and end of sequences that may contain a highlight, and thus allowing end users to set a sensitivity threshold to the system. In fact in the envisioned use case, where forecast highlights are transmitted to a handheld device, some users may prefer to get only very probable highlights, e.g. to reduce the costs related to video transmission, while other users may prefer to see more actions, accepting false alarms.

Only visual features are used by the system, since audio features may not be always available. The features may be divided in two groups: *compressed domain features*, that are extracted directly from the MPEG video stream:

- Motion vectors: MPEG motion vectors are used to calculate indexes of camera pan and tilt, and an index of motion intensity (see Fig. 1);
- Playfield: YUV color components are used to extract and evaluate the playfield framed.

and uncompressed domain features, that are extracted from images

- Players: players are extracted using previous knowledge of team colors (to improve precision) from uncompressed I frame: the ratio of pixels of the two teams is the cue used by the Bayes networks.
- Playfield lines: playfield lines are extracted from the uncompressed I frame: they are filtered out based on length and orientation.

The ratio of playfield framed allows to classify frames in three types (see Fig. 2): long, medium and close shot. The playfield area framed is classified in three zones, using the histogram of line orientation: left, center and right.

(a) (b)

Fig. 1. *a)*: original frame; *b)*: Motion vectors and average motion vector (long red line)

Evidence and inference are computed for each MPEG GOP (12 frames, i.e every 1/2 second in PAL video standard). If the highlight is predicted in the following 6 seconds (12 GOPs) the video is processed by the Bayesian validation networks. Conditional probabilities are updated every 2 secs. Four networks are used to predict highlights: two networks to predict attack actions (left-right) and two networks to predict placed kicks (left-right).

Fig. 2. *a)*: long; *b)*: medium; *c)*: close shot;

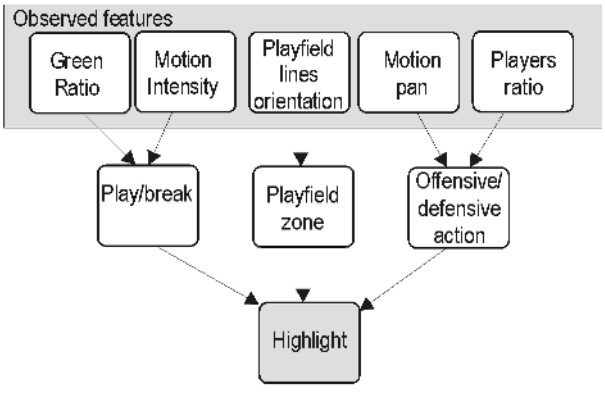


Fig. 3. Bayesian network used for highlight prediction (placed kick and attack action)

Fig. 3 shows the structure of the Bayesian network used for prediction; it is interesting to note that placed kicks are characterized by an initial break phase. The system is able to detect if the predicted action is concluded by a shot on goal. In fact when there is a shot on goal typically there is a sequence composed by 3 phases:

1. Fast panning of main camera toward goal post (Long Shot);
2. Zooming on the player who kicked the ball (Medium Shot or Close Shot);
3. View of the crowd or close up of the trainer (Close Shot).

The sequence is about the same in both cases of a scored goal or of a near miss, and the features that are extracted are the same. It must be noted that due to the soccer rules only the referee can decide if a ball that enters a goal post scores a goal; thus we can simply detect the presence of shot on goal, and not a goal.

To detect the shots on goal two networks are used, one for the left, and one for the right side of the playfield. The networks have the same structure, while the conditional probability tables of the nodes change every 2 seconds following the three typical phases described before. Fig. 4 shows the structure of the Bayesian network used for shot on goal detection.

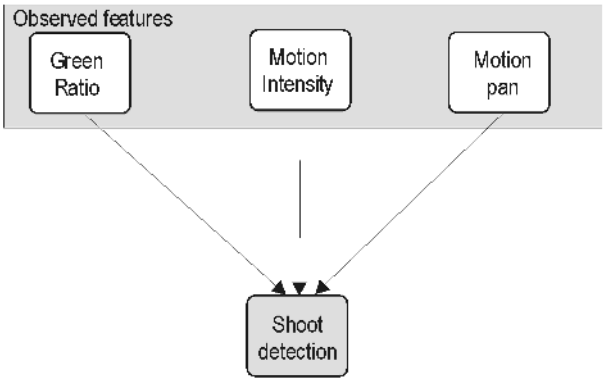


Fig. 4. Bayesian network used for shot on goal detection

The workflow of the system is: feature extraction from P and I frames, feature quantization and the prediction of the Bayes network is evaluated. If the evidence is above the user defined threshold then the shot on goal recognition is activated.

5 Experimental Results

The video stream used for the test set are MPEG-1 and MPEG-2 videos at 25 frames per second (PAL standard) and with a resolution that is respectively of 360×288 and 720×576. The GOP length is 12 frames. 268 case examples (~ 90 min) collected from World Championship 2002 and European Championship 2004 have been used to test the annotation system.

- 172 highlights that have been concluded with a shot on goal (SOG): 134 attack actions (AA) and 38 Placed kicks (PK)
- 54 highlights that have not been concluded with a shot on goal (NSOG): 51 attack actions and 3 Placed kicks
- 42 Other Actions (OA)

Table 1 and 2 report precision and recall figures, and a breakdown of the classification of SOG, NSOG and OA actions, and attack actions and placed kicks. The average number of frames between the prediction and the appearance of a SOG action is 74,2 (~ 3 sec. for a PAL system).

Typically the best results in terms of prediction of SOG are obtained in the case of attack actions or penalty kicks: in the latter case the prediction is performed when the large view of the player that is going to kick is shown. The lower precision of placed kick detection is due to cases of free kicks that are quite far from the goal box area; in this case the area is framed after kicking the ball, and the number of frames between prediction and the actual event is the lowest. Corner kick are less critical because of the typical large view, but since usually prediction starts after the kick the number of frames between prediction and event is quite low. Among the critical attack actions, that cause misses and

misclassifications there the cases in which the attacker does not directly kick the ball toward the goal post, but rather waits or makes small range assists to other team mates.

Analysing table 1, it must be noted that some results of the proposed system, while still not being the expected ones, are still acceptable. E.g. if a SOG is predicted but not recognized is still an acceptable result w.r.t the prediction requirements. The same applies to a NSOG that is predicted and then recognized a SOG. In fact these two types of errors affect only the validation of the forecast.

Table 1. Annotation performance of SOG, NSOG and OA. $\star$: expected behaviour; †: acceptable results; ‡: bad results.

Highlight type	Predicted and SOG recognized	Predicted and SOG not recog.	Not predicted	Precision	Recall
SOG	$151/172^{\star}$	$13/172^{\dagger}$	$8/172^{\ddagger}$	0.96	0.88
NSOG	$7/54^{\dagger}$	$43/54^{\star}$	$4/54^{\ddagger}$	0.74	0.80
OA	$0/42^{\ddagger}$	$2/42^{\dagger}$	$40/42^{\star}$	0.77	0.95
			Avg.	**0.83**	**0.88**

Table 2. Annotation performance of attack actions and placed kicks

Highlight type	Correctly detected	Misclassified/ missed	Precision	Recall
AA	163/185	22/185	0.98	0.88
PK	37/41	4/41	0.63	0.91
		Avg.	**0.83**	**0.88**

6 Conclusions

In this paper we have reported the results of real time annotation system, applied to soccer videos, that forecast the appearance of highlights in real-time, it classifies also the type of highlights and the presence of shots on goal. Our future work will deal with a refinement of the proposed system, extending and specializing the types of highlights that may be forecast, and extending the system to other types of sports.

Acknowledgments

This work has been partially funded by the European VI FP, Network of Excellence DELOS (2004-06). Authors would like to thank Filippo Conforti for his help.

References

1. W. Zhou, A. Vellaikal, and C.C.J. Kuo, "Rule-based video classification system for basketball video indexing," in *ACM Multimedia 2000 workshop*, 2001, pp. 213–126.
2. S. Nepal, U. Srinivasan, and G. Reynolds, "Automatic detection of 'goal' segments in basketball videos," in *Proc. of ACM Multimedia*, 2001, pp. 261–269.
3. S.S. Intille and A.F. Bobick, "Recognizing planned, multi-person action," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 414–445, March 2001.
4. E. Kijak, G. Gravier, L. Oisel, and P. Gros, "Audiovisual integration for tennis broadcast structuring," in *CBMI 2003*, Rennes France, 2003, pp. 421–428.
5. S. Choi, Y. Seo, H. Kim, and K.-S. Hong, "Where are the ball and players? soccer game analysis with color-based tracking and image mosaic," in *Proc. of Int'l Conference on Image Analysis and Processing*, 1997.
6. Y. Gong, L.T. Sin, C.H. Chuan, H. Zhang, and M. Sakauchi, "Automatic parsing of tv soccer programs," in *Proc. of IEEE Int'l Conference on Multimedia Computing and Systems*, 1995, pp. 15–18.
7. R. Leonardi and P. Migliorati, "Semantic indexing of multimedia documents," *IEEE Multimedia*, vol. 9, no. 2, pp. 44–51, April-June 2002.
8. J. Assfalg, M. Bertini, A. Del Bimbo, W. Nunziati, and P. Pala, "Soccer highlights detection and recognition using hmms," in *Proc. of Int'l Conf. on Multimedia and Expo (ICME2002)*, 2002.
9. A. Ekin, A. Murat Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Transactions on Image Processing*, vol. 12, no. 7, pp. 796–807, July 2003.
10. J. Assfalg, M. Bertini, C. Colombo, A. Del Bimbo, and W. Nunziati, "Semantic annotation of soccer videos: automatic highlights identification," *Computer Vision and Image Understanding*, vol. 92, no. 2-3, pp. 285–305, November-December 2003.
11. X.Yu, C. Xu, H.W. Leung, Q. Tian, Q. Tang, and K. W. Wan, "Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video," in *ACM Multimedia 2003*, Berkeley, CA (USA), 4-6 Nov. 2003 2003, vol. 3, pp. 11–20.
12. M. Petkovic, V. Mihajlovic, and W. Jonker, "Multi-modal extraction of highlights from tv formula 1 programs," in *Proc. of IEEE Int'l Conference on Multimedia & Expo*, 2002.
13. F. V. Jensen, *Bayesian Networks and Decision Graphs*, Springer, 2001.

Lightweight Protection of Visual Data Using High-Dimensional Wavelet Parametrization*

Andreas Pommer and Andreas Uhl

Salzburg University, Department of Scientific Computing,
Jakob Haringer-Str. 2, A-5020 Salzburg, Austria
{apommer, uhl}@cosy.sbg.ac.at
http://www.scicomp.sbg.ac.at/

Abstract. A lightweight encryption scheme for visual data based on wavelet filter parametrization is discussed. Being a special variant of header encryption, the technique has an extremely low computational demand. Security assessement of low-dimensional parametrizations schemes show severe weaknesses. We show that using high-dimensional parametrizations the scheme may be employed in applications requiring a medium security level.

1 Introduction

Encryption schemes for multimedia data need to be specifically designed to protect multimedia content and fulfil the security requirements for a particular multimedia application. For example, real-time encryption of an entire video stream using classical ciphers requires heavy computation due to the large amounts of data involved, but many multimedia applications require security on a much lower level (e.g. TV news broadcasting [5]). In this context, several selective encryption schemes have been proposed recently which do not strive for maximum security, but trade off security for computational complexity (see [12] for a comprehensive overview). For example, we mention selective encryption of MPEG streams [1] and of JPEG 2000 data [2, 7]. However, in case that one or more parties which are involved in an application have strong limits on their processing capacities (e.g., a mobile device with a small battery and a slow processor), even encrypting a small fraction of the image data may still be out of reach. In such environments, confidentiality may be provided by an extreme case of selective encryption as will be described in the following.

In recent work, we have proposed lightweight encryption schemes based on orthogonal [3] and biorthogonal [11] wavelet filter parametrizations for applications requiring a low to medium security level. In this work we investigate higher-dimensional wavelet parametrizations in order to increase the key-space and attack resistance of the scheme. In Section 2, we shortly review wavelet

* This work has been partially supported by the Austrian Science Fund FWF, project no. P15170.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 645–652, 2005.

compression and the wavelet filter parametrization scheme in use. Section 3 introduces the encryption scheme, evaluates the resulting compression quality and security, and discusses the use of higher-dimensional parametrizations.

2 Wavelet Compression and Filter Parametrization

Wavelet-based still image compression has to be considered state of the art nowadays, especially in applications requiring low bit rates and bitstream scalability. In the area of standardization the two most prominent techniques are JPEG 2000 and MPEG-4 VTC. The SMAWZ codec [4] used in our experiments is a variant of the well known SPIHT algorithm which has been optimized for efficient implementation using bitplanes instead of lists. In all these compression schemes, filters especially tuned for that specific purpose are employed. However, there exists an almost infinite richness of different wavelet filters to choose from.

For the construction of compactly supported orthonormal wavelets, solutions for the dilation equation have to be derived, satisfying two conditions on the coefficients c_k ($\phi(t) = \sum_{k\in\mathbb{Z}} c_k \phi(2t-k)$, with $c_k \in \mathbb{R}$). In our work we use a family of parameterized filters generated according to an algorithm proposed by Scheid and Pittner [10]:

Given N parameter values $-\pi \leq \alpha_i < \pi$, $0 \leq i < N$, the following recursion

$$c_0^0 = \tfrac{1}{\sqrt{2}} \text{ and } c_1^0 = \tfrac{1}{\sqrt{2}}$$

$$c_k^n = \tfrac{1}{2}\Big((c_{k-2}^{n-1} + c_k^{n-1}) \cdot (1+\cos\alpha_{n-1}) + (c_{2(n+1)-k-1}^{n-1} - c_{2(n+1)-k-3}^{n-1})(-1)^k \sin\alpha_{n-1}\Big)$$

can be used to determine the filter coefficients c_k^N, $0 \leq k < 2N+2$. We set $c_k = 0$ for $k < 0$ and $k \geq 2N+2$. Example filters which can be generated using this formula are the Daubechies-6 filter, which can be constructed using the parameters (0.6830127, −0.1830127), or the Haar filter which is generated with the parameter 0.

Note that the number N of parameter values α_i is denoted as the dimensionality of the parametrization scheme. Larger N lead to longer wavelet filters.

3 A Lightweight Encryption Scheme

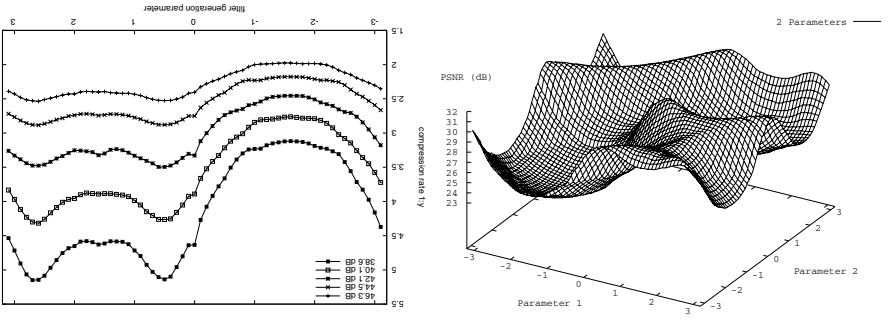
As we have seen, wavelet-based compression can be performed using a wide variety of different wavelet transforms. This degree of freedom may be exploited to add security to wavelet-based applications by only encrypting the header information defining the wavelet transform in use and keeping the rest in plaintext. Following this general idea, selective encryption schemes based on encrypting the secret wavelet packet subband structure [9] or NSMRA decomposition scheme [8] have been proposed recently.

Besides their use in encryption, secret wavelet filters generated by filter parametrizations like those reviewed in section 2 have been proven to increase security in wavelet-based watermarking schemes [6]. In this paper we investigate the properties of a header encryption variant where we keep the parameter to generate the filters for the wavelet transform secret. For example, this can be easily achieved in the context of JPEG 2000 Part II by simply encrypting the corresponding field containing the user-defined custom filters in the header using a cryptographically strong cipher. As a consequence, the amount of data subject to encryption is minimal, since no actual image data but only filter coefficients are encrypted.

In the following subsections, we investigate the compression quality and the security of the resulting scheme.

3.1 Compression Quality

Whereas the traditional filters used for wavelet compression are tuned for optimal concentration of the energy contained in the image and the separation of high- and low-frequency parts, parameterized filters provide a wide quality range. The advantage as well as the disadvantage of parameterized filters is their variety, not all filters within such a family are equally suited for a specific purpose like image compression. Fig. 1(a) shows the resulting compression ratios when compressing the 8 bpp 512 × 512 pixels Lena image using different parameter values to a set of fixed quality levels in the range between 38 and 46 dB PSNR.

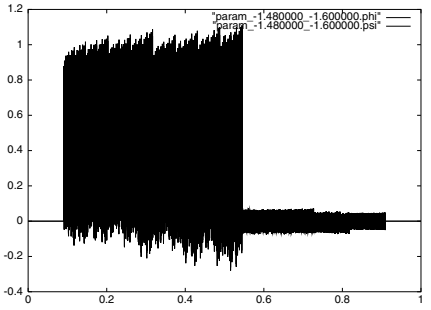


(a) Compression rate for various quality levels, 1-d parameterization

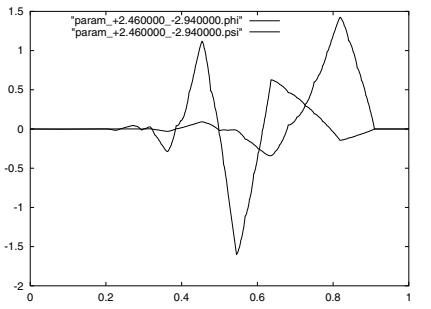
(b) Compression quality at fixed bitrate, 2-d parameter space

Fig. 1. Compression results employing parametrized filters within SMAWZ.

It is clearly displayed that the file sizes obtained by the filters resulting from the parametrization algorithm described in Section 2 vary to a large extent. Obviously, the differences increase when decreasing the bitrate. Among other (smaller) variations, the left half of the parameter range leads to poor filter quality. In Fig. 1(b) we display the PSNR quality when compressing the Lena



(a) 2-parameter filter giving poor compression results



(b) 2-parameter filter giving good compression results

Fig. 2. Filters generated by 2 parameters

image to 40000 bits (the same bitrate is used in all subsequent experiments) with different filters generated by the parametrization scheme using 2 parameters.

Similar behaviour with respect to quality variations can be observed in the two-dimensional case. Again the quality varies in a wide range, here we observe a minimum of 23.4 dB (generated by the filter shown in Fig. 2(a)) and a maximum of 32 dB (for the corresponding filter see Fig. 2(b)), with an average of 27.7 dB.

As a consequence of these findings, a strategy is required to limit the possible loss in compression quality introduced by randomly chosen parameters. The most desirable approach would be a heuristic which – given either the parameters to generate the filters or the actual filter coefficients themselves – could determine an approximation of the compression quality to be expected in advance (i.e. without performing the actual compression). Unfortunately, besides restricting the parameter to positive values in the one-dimensional case, no such heuristic could be found.

Generating the parameters and performing the actual compression stage to determine the corresponding quality is too time consuming (since only one failure in parameter choice (i.e. one bad quality filter) makes the scheme significantly more expensive than a full AES encryption of a classically encoded bitstream). Therefore, we determine parameter values of good quality in advance and restrict the admissible parameters to regions close to that values. Fortunately, the quality of parameters is very much image independent, which makes this approach a feasible and efficient one. However, the decrease of the amount of admissible parameter values is known in advance (also to a potential attacker) and reduces the overall security of the system since it corresponds to a smaller keyspace.

3.2 Security

The data type of the parameters is $\mathbb{R}$ (in theory), in practice it is $\mathbb{Q}$ which means we need to discretize the parameter space applying a fixed size grid onto it. Close parameters lead to similar filters which in turn lead to similar wavelet transform coefficients. Of course, this might be a threat to the security of the system since

an attacker does not need to know the compression parameter exactly to get a "decrypted" image with sufficient quality. Therefore, the discretization (i.e. the grid size) needs to be defined in a way that different parameters lead to sufficiently different filters.

In Fig. 3 we illustrate this problem. The Lena image is compressed with filters generated by six parameters, and subsequently decompressed with a large number of different filters derived from parameters covering the range $\pm\pi$ centered at the "correct" parameter. We plot the PSNR of the resulting images against the parameter used for decompression. The desired result would be an isolated single PSNR peak at the position of the "correct" parameter (that one used for compression) and low values everywhere else.

The result of this experiment is not an isolated PSNR peak but an entire region centered around the correct parameter where the PSNR values are decreasing with increasing distance from the correct value. In Fig. 4 we visualize images where the Lena image was compressed using the parameter 1 and decompressed with parameters displaced from the correct one by a certain amount.

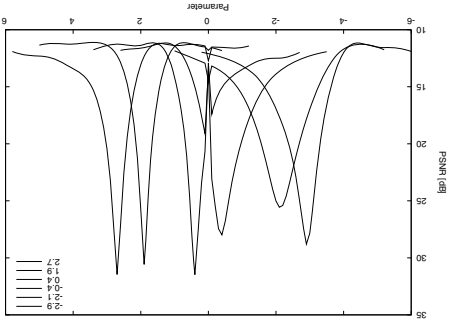
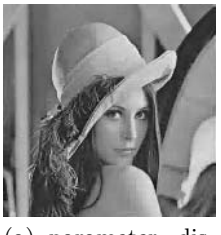


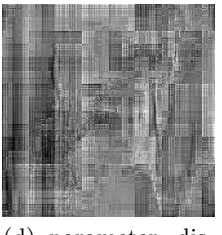
Fig. 3. Attack against a 1-D parameter scheme



(a) parameter distance of 0

(b) parameter distance of 0.2

(c) parameter distance of 1.0



(d) parameter distance of 3.1

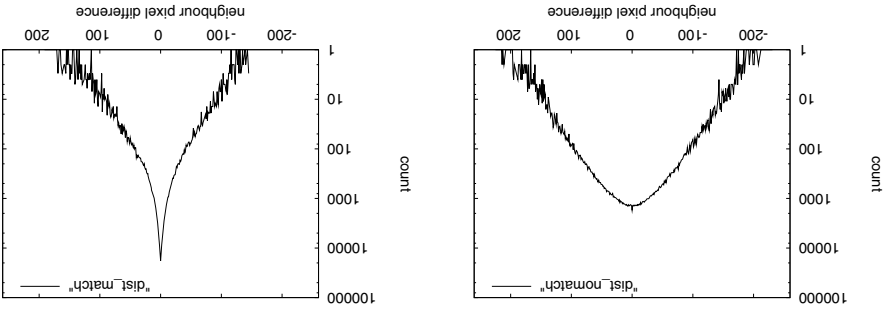
Fig. 4. Images resulting from compression and decompression with similar parameterized filters

Obviously, the quality of the image in Fig. 4(b) is too high to provide any kind of confidentiality (compression and decompression paramters are too close), whereas the quality of Figs. 4(c) and 4(b) is low enough for applications requiring a low to medium confidentiality level.

As a consequence, the number of admissible parameter values needs to be restricted to a rather sparse discretization grid. Taken this fact together with the beforementioned restrictions due to low quality filters (subsection 3.1), of course the keyspace is too small for a reasonable application in case of the 1-D

parameter scheme. However, higher dimensional parametrizations can provide a sufficiently large amount of different parameters (see next subsection).

An attacker wanting to recreate the image without knowledge of the correct parameters (and without the plaintext image) has to test a grid spanning the search space of possible parameters and tries to deduce information about the correct parameters. In order to do this, the (possibly only partly) reconstructed images need to be automatically evaluated with respect to their "perceptual quality". One possibility to achieve this is to exploit the fact that incorrect filters generate more high-frequency noise. A simple technique in this context is to compute the differences between neighbouring pixels of image reconstructions.



(a) matching filter (b) non-matching filter

Fig. 5. Statistical distribution of pixel differences for matching and non-matching filters in the 1-dimensional case

Fig. 5 shows histograms of the magnitude of the pixel differences, Fig. 5.a in case the correct filter is used for reconstruction, Fig. 5.b in case the filter used for reconstruction is far away from the correct one. In the matching case the amount of low differences is higher, and high differences do not exist, in the non-matching case the opposite holds. For automated search the distance to an histogram like that in Fig. 5.b can be minimized using a gradient technique, for example.

3.3 Higher-Dimensional Parametrizations

A possible strategy to increase the available keyspace significantly is to move to schemes with more parameters (leading to longer filters). Of course, the problem of varying filter qualities also exists in higher dimensions. To quantify the corresponding properties, we randomly select parameters and perform the compression with the same fixed bit limit as before and record for each number of parameters N the minimal, average, and maximal PSNR. Table 1 shows evidence that the maximum PSNR slightly increases, whereas the minimum and average values significanlty decrease for an increasing number of dimensions N. This result shows that for larger N a strategy to avoid low quality filters is even more important.

Table 1. Quality (PSNR in dB) tests with random parameters

#param	#samples	min	avg	max
1	25	24.94	28.36	31.50
2	69	23.40	27.49	31.44
3	193	22.22	26.84	31.99
4	603	21.67	26.37	32.14
5	1875	20.59	25.87	32.10
6	5470	20.14	25.57	32.07
7	8766	20.16	25.26	32.16
8	13185	19.88	24.98	32.09

On the other hand, the results available so far do not guarantee the existence of an increasing number of high quality filters for increasing dimension N. In Fig. 6 we plot the empirical cumulative distribution function of the filters with parameter dimensions $N = 1, 2, \ldots, 8$ with respect to their compression quality. The relative number of high quality filters obviously decreases with increasing N. For example, we see that for $N = 1$, about 45% of all filters show quality ≥ 30 dB, for $N = 2$ about 20%, and for $N = 8$ approximately 2% of all filters exhibit quality ≥ 30 dB.

However, considering the exponential growth of the overall number of available filters with respect to the dimension of the parametrization scheme (assuming a constant, dimension-independent discretization of the parameter range), a significant increase of the absolute number of high quality filters is guaranteed when increasing N.

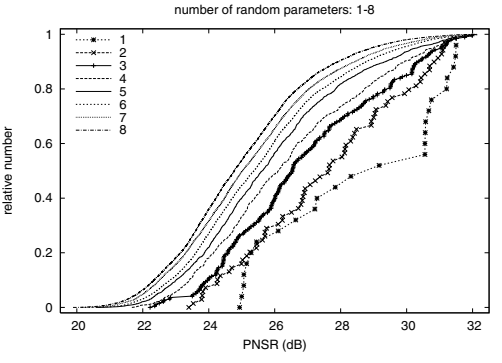


Fig. 6. Relative number of good and poor quality filters

But does the larger keyspace help to avoid an attack as described in the previous subsection? An exhaustive search through the parameter space is simply too costly, and varying all parameters simultanously during a random search does not enable a steepest descent search technique due to the high dimensionality. Once a single parameter combination with promising smoothness properties is found, it is not clear how to further improve the result. On the other hand, fixing all parameters but that in one dimension does not lead to clearly improved smoothness behaviour once that single parameter is near to its optimum. Therefore, a separable search (i.e. optimizing each parameter separately) is not successful as well. It turns out that all considered automated techniques fail to reliably identify an approximation to the correct parameter set for larger N due to the size of the key-space and the increasing non-linearity of the search space with increasing dimension.

4 Conclusions and Future Work

We have shown that higher dimensional wavelet filter parametrizations may solve the security problems of recently proposed lightweight encryption schemes for visual data. As the compression performance of the orthogonal filters in use is inferior to the standard biorthogonal filters employed within codecs, we will focus in future work on parametrization techniques directly related to the lifting scheme to integrate the approach into JPEG 2000.

References

[1] B. Bhargava, C. Shi, and Y. Wang. MPEG video encryption algorithms. *Multimedia Tools and Applications*, 24(1):57–79, 2004.

[2] Raphaël Grosbois, Pierre Gerbelot, and Touradj Ebrahimi. Authentication and access control in the JPEG 2000 compressed domain. In A.G. Tescher, editor, *Applications of Digital Image Processing XXIV*, volume 4472 of *Proceedings of SPIE*, pages 95–104, San Diego, CA, USA, July 2001.

[3] T. Köckerbauer, M. Kumar, and A. Uhl. Lightweight JPEG 2000 confidentiality for mobile environments. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '04*, Taipei, Taiwan, June 2004.

[4] R. Kutil. A significance map based adaptive wavelet zerotree codec (SMAWZ). In S. Panchanathan, V. Bove, and S.I. Sudharsanan, editors, *Media Processors 2002*, volume 4674 of *SPIE Proceedings*, pages 61–71, January 2002.

[5] Benoit M. Macq and Jean-Jacques Quisquater. Cryptology for digital TV broadcasting. *Proceedings of the IEEE*, 83(6):944–957, June 1995.

[6] P. Meerwald and A. Uhl. Watermark security via wavelet filter parametrization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'01)*, volume 3, pages 1027–1030, Thessaloniki, Greece, October 2001. IEEE Signal Processing Society.

[7] Roland Norcen and Andreas Uhl. Selective encryption of the JPEG2000 bitstream. In A. Lioy and D. Mazzocchi, editors, *Communications and Multimedia Security. Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, CMS '03*, volume 2828 of *Lecture Notes on Computer Science*, pages 194 – 204, Turin, Italy, October 2003. Springer-Verlag.

[8] A. Pommer and A. Uhl. Wavelet packet methods for multimedia compression and encryption. In *Proceedings of the 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 1–4, Victoria, Canada, August 2001. IEEE Signal Processing Society.

[9] A. Pommer and A. Uhl. Selective encryption of wavelet-packet encoded image data — efficiency and security. *ACM Multimedia Systems (Special issue on Multimedia Security)*, 9(3):279–287, 2003.

[10] J. Schneid and S. Pittner. On the parametrization of the coefficients of dilation equations for compactly supported wavelets. *Computing*, 51:165–173, May 1993.

[11] A. Uhl and A. Pommer. Are parameterised biorthogonal wavelet filters suited (better) for selective encryption? In Jana Dittmann and Jessica Fridrich, editors, *Multimedia and Security Workshop 2004*, pages 100–106, Magdeburg, Germany, September 2004.

[12] A. Uhl and A. Pommer. *Image and Video Encryption. From Digital Rights Management to Secured Personal Communication*, volume 15 of *Advances in Information Security*. Springer-Verlag, 2005.

Shot Detection and Motion Analysis for Automatic MPEG-7 Annotation of Sports Videos

Giovanni Tardini, Costantino Grana, Rossano Marchi, and Rita Cucchiara

Department of Information Engineering, University of Modena and Reggio Emilia,
Via Vignolese 905/b, 41100 Modena, Italy,
{surname.name}@unimore.it

Abstract. In this paper we describe general algorithms that are devised for MPEG-7 automatic annotation of Formula 1 videos, and in particular for camera-car shots detection. We employed a shot detection algorithm suitable for cuts and linear transitions detection, which is able to precisely detect both the transition's center and length. Statistical features based on MPEG motion compensation vectors are then employed to provide motion characterization, using a subset of the motion types defined in MPEG-7, and shot type classification. Results on shot detection and classification are provided.

1 Introduction

Video annotation is one of the primary processes in the life cycle of multimedia digital libraries. Automatic annotation must provide a description of the meaningful parts of the video in a standard way to be suitable used in further accessing and content based retrieval processes. In this framework, the MPEG-7 standard is becoming very popular.

Annotation must cope with an available ontology of concepts endowed in the video that are often defined by the users of digital libraries. In sports video the ontology can be easily defined being the rules of the play and the appearance of the video predictable and with periodical occurrence.

In this paper we describe general algorithms that are devised for MPEG-7 automatic annotation of Formula 1 videos, and in particular for *camera-car* shots detection. This is a challenging task for many reasons: strong camera motion as well as objects motion is present, and color features are not always important cues for shot classification, since they discriminate between cars and teams but not between events.

Despite of the very specific application, the algorithms here proposed are very general and can be applied in many contexts where shots with a specific motion type (e.g. zooms) must be detected.

In this work we followed the standard approach for edited video annotation that consists in an initial shot detection step and then in shot classification according with certain visual features. We employed a shot detection algorithm suitable both for cuts and linear transitions detection, which, unlike other approaches in literature, is able to precisely detect both the transition's center and length. Then, motion features are extracted to provide a proper classification and motion characterization of shots.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 653 – 660, 2005.

2 Shot Detection

The first step in edited video analysis and characterization is shot detection. In recent years most techniques concentrated on the compressed domain [1]. These works, to obtain faster analysis, employ only information directly available from the MPEG streams, but comparative studies have demonstrated that they perform much worse on gradual transitions [2]. This is especially true in presence of scene with strong motion. Therefore, latest research on video segmentation is particularly focused on gradual transitions.

In [3] a linear transition model is exploited, but the author doesn't deal with the choice of the length of the window. A more refined approach is proposed in [4], where authors deal with long transitions, while in [5] the author expose a comparative study of most of the metrics used in shot detection approaches, both in compressed and uncompressed domain and then proposes an algorithm to detect both abrupt and gradual transitions, with an algorithm whose performances are strictly dependent on the search window length. The authors of [6] describe a neural network approach, trained with a dissolve synthesizer. The classifier detects possible dissolves at multiple temporal scales, and merges the results with a winner-take-all strategy. The algorithm works on contrast-based features, as well as color-based features, and has given good result compared to standard approaches based on edge change ratio.

Before describing our algorithm, it's useful to stress some properties of linear transitions. Let's consider two shots in a video sequence, the first one ending at frame *e*, and the second one starting at frame *s*, with $e < s$, between which a transition occurs. Two hypotheses are made: the first one is that a feature $F(n)$ exists for each frame I_n, with the characteristic of being discriminating and constant for each shot; the second hypothesis is that the transition is linear and *L* frames long, where $L = s - e + 1$. This model includes abrupt cuts too, as transitions with length $L = 0$.

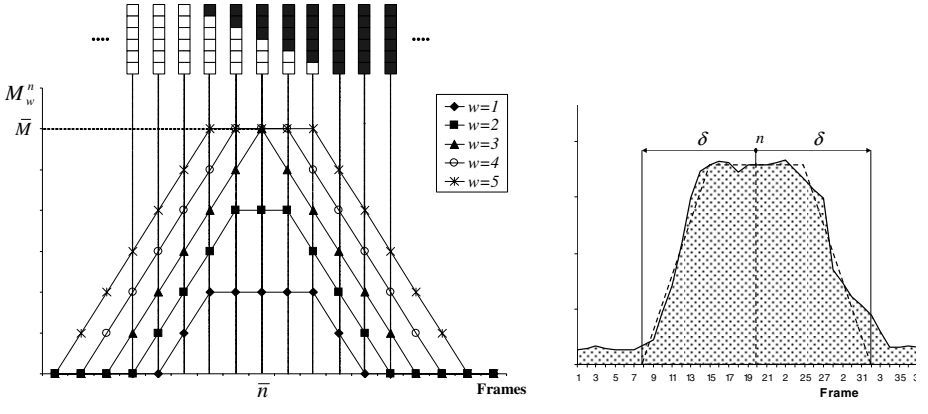


Fig. 1. (a) Values of M_w^n for an ideal linear transition with $L = 5$ at varying *w*. (b) Values of M_w^n in a real case with a chosen *w* value

The transition center is defined as $\bar{n} = (e+s)/2$ and may correspond to a non integer value, that is an inter-frame position. In order to detect the linear transition, we define a function M_w^n which is a difference measure for the feature *F*. This function is computed with sub-frame granularity and is centered on frame or half-frame *n*, with $2n \in \mathbb{N}$, and with window size $2w \in \mathbb{N}$.

In Fig. 1 we see an example of an ideal linear transition with L=5, from a white to a black image. If the transition is linear according with the previous hypotheses, the shape of function *M* is an isosceles trapezoid, centered in $\bar{n}$, that degenerates into a triangle when $2w = L+1$. It's easy to verify that, given the model and M_w^n, each side extends over $\min(2w, L+1)$ frames, and the minor base is $|2w-(L+1)|$ long. It's also possible to show that:

$$\begin{aligned} M_w^{\bar{n}} &< \bar{M} \quad \text{if } 2w < L+1 \\ M_w^{\bar{n}} &= \bar{M} \quad \text{if } 2w \geq L+1 \end{aligned} \tag{1}$$

where $\bar{M} = \max_{w,n} M_w^n$. We define $\psi_{w,L}^n(i)$ the ideal trapezoidal shaped function with the described properties. In the real case, it's not possible to obtain an ideal trapezoid from the data, so we have to look for the parameters that provide the best matching between data and the ideal $\psi_{w,L}^n(i)$ function. To solve this optimization problem, we employ an algorithm constructed of two iteratively repeated steps: the first one searches for the center position *n* and transition length *L*, assuming a fixed window size 2*w*, which is then optimized by the second step, exploiting the property of Eq. 1. The two steps are iteratively repeated, progressively decreasing the estimate error.

The first step begins with a small window $\bar{w}$, and the best trapezoid is searched moving the center *n*, and trying different values for *L*. The trapezoid extends over $\delta = \min(2w, L+1) + |w-(L+1)/2|$ frames on the left and on the right of the center frame. For each couple of *n* and *L* the following measure is computed:

$$\Phi_{\bar{w},L}^n = \sum_{i=n-\delta}^{n+\delta} \min\left(M_{\bar{w}}^i, \psi_{\bar{w},L}^n(i)\right) - \sum_{i=n-\delta}^{n+\delta} \left| M_{\bar{w}}^i - \psi_{\bar{w},L}^n(i) \right| \tag{2}$$

After finding the trapezoid which maximizes $\Phi_{\bar{w},L}^n$, we consider $\bar{n}$ the candidate transition center. Observing Fig. **1**, the value of M_w^n in the ideal case linearly grows with the window *w*, up to the window corresponding to $w = (L+1)/2$ and successively it is stable, leading to a horizontal straight line in the graph.

We employ this property in the second step of the algorithm to give a different estimate of the transition length by finding the smallest window 2*w* that maximizes M_w^n. To provide a more robust technique for the real case, the tilt change of the graph is searched by optimizing the function:

$$Z_w^{\bar{n}} = \sum_{i=0}^{w} \left| M_i^{\bar{n}} - \frac{M_w^{\bar{n}}}{w} i \right| + \sum_{i=w+1}^{W} \left| M_i^{\bar{n}} - M_w^{\bar{n}} \right| \tag{3}$$

where W is the maximum size that a transition can assume. The w value that minimizes $Z_w^{\bar{n}}$ is then used for the next iteration of the first step.

Given the transition length L and its center $\bar{n}$, as detected by the algorithm, we must verify how much the real data fit to the linear transition model. To this aim, we define an error measure as

$$err_{\bar{w}}^{\bar{n}} = \frac{1}{4\bar{w}+1} \sum_{i=-2\bar{w}}^{2\bar{w}} \left| M_{\bar{w}}^{\bar{n}+i} - \psi_{\bar{w},L}^{\bar{n}}(\bar{n}+i) \right| . \tag{4}$$

Here we assume $L = 2w-1$, which causes $\psi_{w,L}^{n}(i)$ to become a triangular shaped function. The error sum is divided by the triangle's base to obtain a measure independent from the transition length. We also introduce the ratio

$$r_{\bar{w}}^{\bar{n}} = \frac{Peak_{\bar{w}}^{\bar{n}}}{err_{\bar{w}}^{\bar{n}}}, \; Peak_{\bar{w}}^{\bar{n}} = M_{\bar{w}}^{\bar{n}} - \min\left(M_{\bar{w}}^{\bar{n}-2\bar{w}}, M_{\bar{w}}^{\bar{n}+2\bar{w}} \right). \tag{5}$$

The Peak value measures the height of the center value with respect to the lower of the two values of M in correspondence to the extremes of the triangle, and provides information on the transition significance, while the ratio provides a normalized estimate of the sequence similarity with a linear transition. These two values are employed to discriminate true and false transitions.

The algorithm could be applied to every frame of the analyzed video sequence, but it is computationally quite expensive. Thus, we employ a fast pre-processing algorithm to discard frames which are very unlikely to be part of a transition. In our experiments, we used the linear discriminant analysis on MPEG features (DC coefficients, number of intra, forward, backward, and interpolated macro-blocks) and set the threshold to obtain a very high recall with lower precision rates.

3 Motion Characterization

For content analysis, several approaches were developed for camera motion characterization in the spatial domain, and some used MPEG motion vectors as an alternative to optical flow [7]. Akutsu et al. [8] presented a camera operation recognition method based on the analysis of motion vector fields by matching motion vector fields with predefined models in Hough space for different types of camera operations. A method to analyze the optical flow in a decomposed manner (projected x and y components were used) was proposed by [9].

A robust statistical method with a six-parameter affine motion model was developed by Bouthemy et al. [10] to detect shot change and camera motion. The use of a three-parameter motion model and a least-squares technique for estimation of camera operation was proposed in [11]. In [12], the spatiotemporal image sequence is constructed by arranging each frame close to the other and forming a parallelepiped with time being the third dimension. Camera operations are recognized by texture analysis of the different faces with the 2D discrete Fourier transform.

It has been verified in literature that MPEG motion compensation vectors are suitable for motion analysis applications, and allow a fast analysis of the basic motion

properties of frames. In [13], for example a rough image subdivision in quadrants was employed to perform image queries on videos. Similarly, we chose to provide a frame level motion description by dividing the image in 4 quadrants, and for each one we computed three motion features based on the average motion vector: its magnitude, its angle and the deviation of motion vectors from the average:

$$\Delta^i = \sqrt{\sum_{j \in MV^i} \left(\min\left(\left| \alpha_j^i - \bar{\alpha}^i \right|, 2\pi - \left| \alpha_j^i - \bar{\alpha}^i \right| \right) \right)^2} \tag{6}$$

where, with respect to quadrant i, MV^i is the set of motion vectors, α_j^i is the direction of the j^{th} motion vector and $\bar{\alpha}^i$ is the direction of their average. Extra care must be taken when using motion vectors. In MPEG video frames, boundary blocks and large smooth areas are most likely to have erroneous motion vectors. Usually, some kind of morphological or median filter should be applied to the motion vector field to remove outliers, before they are used for analysis and indexing. We exploit the Extended Vector Median (EVM), as in [14], to filter the motion vectors in a 5x5 window, that is:

$$mv_{EVM} = \begin{cases} mv_{AVE} = \dfrac{1}{N} \sum_{i=1}^{N} mv_i & \text{if } \sum_{i=1}^{N} \left\| mv_{AVE} - mv_i \right\| \le \sum_{i=1}^{N} \left\| mv_{VM} - mv_i \right\| \\ mv_{VM} & \text{otherwise} \end{cases}, \tag{7}$$

with

$$mv_{VM} \in MV : \sum_{i=1}^{N} \left\| mv_{VM} - mv_i \right\| \le \sum_{i=1}^{N} \left\| mv_j - mv_i \right\|, \forall mv_j \in MV \ . \tag{8}$$

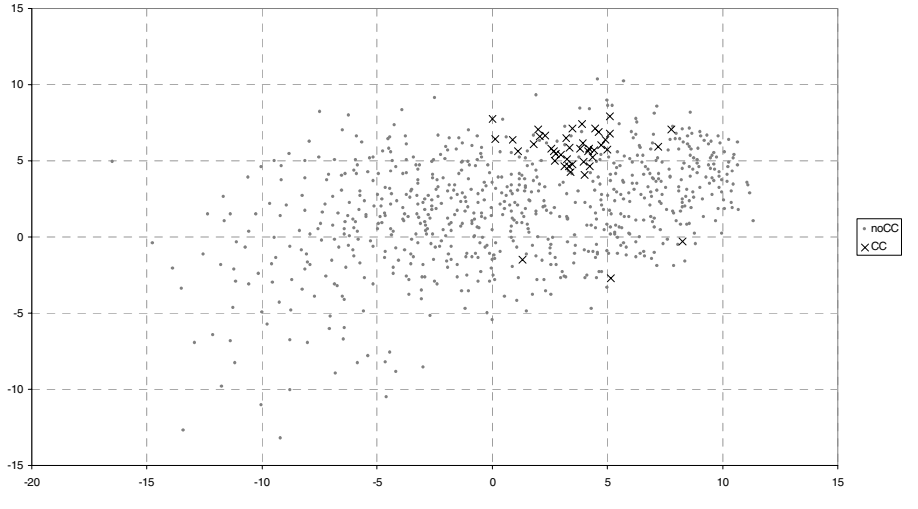


Fig. 2. Sammon's mapping of 755 classified shots

Even if this choice is not of immediate understanding, it has proved to be definitely much more robust than the simple median to situations in which poor motion vectors are available, and the directions are scattered all around. Conversely, it is not misled by single spurious vectors in case of well defined structured motions. With this approach we extract 3 features per quadrant, that is, a total of 12 features per frame.

Motion features thus extracted are exploited to provide both classification and motion characterization of the shots: the classification is based upon the type of camera from which the shot is taken, i.e. camera-car or PTZ camera, while motion characterization is constituted by statistics on the prominent motion types in the shots.

For the classification task, we just aggregated the 12 motion features of all the frames within the shot, by taking the mean feature vector and the standard deviation of every feature. This quite simple summary provides information on the shot motion main characteristics, but also allows describing if the observed motion is consistent along the whole duration of the shot. In Fig. **2**, the 24 dimensional space of an entire Formula 1 video is shown, after projection into two dimensions by means of Sammon's mapping, that is a Non Linear Projection (NLP) procedure for data visualization, which tries to preserve the inter point distances [15]. It is possible to observe that, even if some outliers are present, the Camera Car data clusters together, reassuring on the descriptive power of the motion features chosen. Linear Discriminant Analysis (LDA) was employed to produce a separating hyper plane for the Camera Car shot description.

For the motion characterization task, we considered the five most represented camera movements in Formula 1 videos, which are, employing MPEG-7 terminology, *Fixed* (no motion), *Pan Left, Pan Right, Zoom In,* and *Zoom Out*. We disregarded *Tilt*, *Rotate* and all the camera position movements (*Track, Boom* and *Dolly*) because these represent very rarely observed events and do not appear alone, that is we always observe them in combination with other movements. Since, differently from before, no simple clustering was evident in NLP, we classified the camera movements by k-NN (with k=3) using a set of graphically selected prototypes. Each shot is annotated with the percentages of the different motion types (by taking into account all the frames constituting the shot except the transitions) and their respective amounts of motion, computed with the means of motion vectors' intensities of the quadrants.

4 MPEG-7 Stream Description

The content of the analyzed video clip is described using the MPEG-7 standard. In Fig. 3 an example of the output is shown. The video is divided in a set of instances of *VideoSegmentType* Descriptor Schema (DS), one for each shot detected by our linear transition detection algorithm. The start of each shot (after the transition's end) is located in the file by its position in bytes (*BytePosition* local type) and a reference in time (*MediaTimeType* DS), expressed in number of frames at a specific frame rate (in the example 25 fps). Each shot has a duration equal to the shot's length excluding the transitions at the beginning and at the end of it.

The type of camera used in the shot is referred from the cameras classification scheme as a semantic descriptor (*SemanticType* DS).

Fig. 3. (a) Example of the MPEG-7 annotation produced by our system. (b) Example of camera car shot. (c) Example of PTZ camera shot

The MPEG-7 descriptor schema for camera motion (*CameraMotionType* DS) is briefly defined as follows: for each of the possible camera movements (Pan, Tilt, Zoom, Roll, etc.) both a fractional presence and an amount of motion can be specified. The first is the fraction of the total duration for which a certain motion type is present, and the second describes the average speed of the motion type. We used a single camera motion description for each video segment, which overlaps with the entire segment, but multiple descriptors for each segment could be used as well.

Using MPEG-7 video content description allows our indexes to be compatible with other annotation tools such, as IBM's VideoAnnex [16], and to have a description which can be expanded in a later time without compatibility problems.

5 Results

During the shot detection algorithm development, we used a series of Formula 1 selected sequences as training examples for the choice of both the thresholds and the histogram and spatial metrics linear combination coefficients. The tests here described are evaluated against a ground truth dataset composed of about 125.000 training frames and 160.000 test frames, from a Formula 1 TV videos Digital Library. Here, the pre-processing algorithm selected 1754 possible transitions, including 175 real linear transitions and 539 cuts. Within this dataset, for gradual transitions our algorithm reached a 90% of recall and 82% precision, while for abrupt cut the obtained values are 97% and 90%.

Shot classification was later applied to the obtained segmentation to identify Camera Car instances. Cross-validated LDA provided 88.4% recall and 80.9%

precision. Higher recall rates can be set, depending on the penalty assigned to false detections: for example in our case we missed 5 over 43 Camera Car shots, but we could get 4 more correctly classified at the price of 21 more false positives. The users feeling was that it was better to have more false positives than to have to search manually in the whole video a Camera Car shot they knew it was there.

The further motion characterization allowed interesting and often satisfactory query capabilities and similarity searches (e.g. one could query all the shots with more than 60% of panning), but accurate analysis on the test set are currently being performed.

Acknowledgements

The project is funded by the European Network of Excellence DELOS of the VI Framework Program. We thank Ferrari S.p.A. for the video database availability.

References

1. Pei, S.-C., Chou, Y.-Z.: Efficient MPEG Compressed Video Analysis Using Macroblock Type Information. IEEE Trans. Multimedia 1 (1999) 321–333
2. Gargi, U., Kasturi, R., Strayer, S.H.: Performance Characterization of Video-Shot-Change Detection Methods. IEEE Trans. Circuits Syst. Video Technol. 10 (2000) 1–13
3. Yeo, B.-L., Liu, B.: Rapid Scene Analysis on Compressed Video. IEEE Trans. Circuits Syst. Video Technol. 5 (1995) 533–544
4. Heng, W.J., Ngan, K.N.: Long transition analysis for digital video sequences. Circuits Syst. Signal Process. 20 (2001) 113–141
5. Bescos, J.: Real-Time Shot Change Detection Over Online MPEG-2 Video. IEEE Trans. Circuits Syst. Video Technol. 14 (2004) 475–484
6. Lienhart, R., Zaccarin, A.: A System for Reliable Dissolve Detection in Videos. In: Proc. Int. Conf. Image Proc. (2001) 406–409
7. Patel, N.V., Sethi. I.K.: Video shot detection and characterization for video databases. Pattern Recognit. (Special Issue on Multimedia) 30 (1997) 583–592
8. Akutsu, A., Tonomura, Y., Hashimoto, H., Ohba, Y.: Video indexing using motion vectors. In: Proc SPIE (Visual Commun Image Process) 1818 (1992) 1522–1530
9. Xiong, W., Lee, J.C.-M.: Efficient scene change detection and camera motion annotation for video classification. Comput. Vis. Image Underst. 71 (1998) 166–181
10. Bouthemy, P., Gelgon, M., Ganansia, F.: A unified approach to shot change detection and camera motion characterization. IEEE Trans. Circuits Syst. Video Technol. 9 (1999) 1030–1044
11. Milanese, R., Deguillaume, F., Jacot-Descombes, A.: Efficient segmentation and camera motion indexing of compressed video. Real-Time Imaging 5 (1999) 231–241
12. Maeda, J.: Method for extracting camera operations to describe sub-scenes in video sequences. In: Proceedings of IS&T/SPIE conference on digital video compression on personal computers: algorithm and technologies, San Jose, 2187 (1994)56–67
13. Ardizzone, E., La Cascia, M., Avanzato, A., Bruna, A.: Video indexing using MPEG motion compensation vectors. Proc. IEEE Int. Conf. Multimedia Comp. Syst. (1999) 725– 729
14. Astola, J., Haavisto, P., Neuvo, Y.: Vector median filters. Proc. IEEE 78 (1990) 678–689
15. Sammon, Jr. J.W.: A nonlinear mapping for data structure analysis. IEEE Trans. Comput. C-18 (1969) 401–409
16. http://www.research.ibm.com/VideoAnnEx/

Tracking Soccer Ball in TV Broadcast Video

Kyuhyoung Choi and Yongduek Seo

Dept. of Media Technology, Sogang University,
Seoul, Korea 121-742
{kyu, yndk}@sogang.ac.kr
http://vglab.sogang.ac.kr/yndk_htdocs/htdocs/research_tracking.html

Abstract. This paper focuses on soccer ball tracking which is known to be more difficult than that of players due to its small size in an image and abrupt changes in its motion. Suggested is an effective soccer ball tracking algorithm which estimates ball position by exploiting the background image and player tracking results. In other words, the trajectory of ball is derived as image blobs by eliminating player blobs and the background parts from an image sequence. This algorithm performed well on a pretty long TV broadcast sequence in which the ball is frequently occluded by players.

1 Introduction

Soccer has been titled the most popular sport worldwide and even today Worldcup is considered as a global festival with far-reaching effects. Analysis of soccer video sequences has been an interesting application in computer vision and image analysis as more and more related papers are published recently.

Tracking players and ball must be a necessary step before an higher level analysis. There have been some researches on tracking players [1,2,3,4,5,6,7,8,9]. Among them, the papers such as [2,3,9] have dealt with the ball tracking as well.

However ball tracking has not been thoroughly studied yet and that is the focus of this paper. Even though ball tracking belongs to single object tracking while player tracking falls within multi-object tracking, ball tracking is not easier than players tracking due to following aspects. Usually ball blobs in images are very small, which makes it difficult to derive features from and to be characterized. Sudden changes in its motion is another factor to make it challenging. In addition, occlusion and overlapping with players causes a severe problem in tracking the ball continuously; The ball becomes invisible and appears at places where a continuous prediction could not reach. In [10], it is evaluated whether a candidate trajectory, which is generated from the candidate feature image by a candidate verification procedure based on Kalman filter, is a ball trajectory instead of whether a sole object is a ball. In [11], an indirect ball detection strategy based on non-ball elimination is applied and CONDENSATION algorithm, a simple version of particle filters, is used to track ball.

Our approach is based on the work of [12] where an image of ball trajectory blob is derived to be used as the proposal density in particle filtering frame work. The soccer sequences were taken from a fixed camera while ours a moving camera.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 661–668, 2005.

The ball tracking as well as the players tracking in this paper is done by using particle filters, or equivalently, by SMC (Sequential Monte Carlo) methods [13,14,15,16,17]. In tracking multiple blobs of the players, we utilized the method proposed in [5] to address the problem of particle migration during occlusion between the same team players by probabilistic weighting of the likelihood of a particle according to the distance to its neighbors. This paper then concentrates on tracking the ball in a soccer video sequence. We utilize the result of players tracking in order to obtain measurement images that do not have players' blobs.

As mentioned above, two major problems we consider in this paper are 1) the image portion of the ball in a frame is as small as 3×3 in pixels and the color is almost white but blurred due to its motion, and 2) the interaction with players causes overlapping or occlusion and makes it almost impossible to detect and predict the ball area in the sequence by a simple usage of a particle filter.

To solve the first problem, we remove the image blobs of the players using the result of the players' tracking, segment out the ground field using a lower threshold, and finally accumulate the image blobs through the sequence. After an image filtering, this procedure results in a ball blobs connected continuously. Based on this accumulation image, particles are randomly generated only from those areas that have some blobs, which could be a noise blob, too, due to incomplete segmentation. Then, the particle filter evaluates each of the random particles to produce a tracking result.

However, when occlusion or overlapping happens the accumulation does not provide meaningful ball blobs any more. In this case, our tracker changes the ball tracking mode to *invisible* from *visible*, finds and marks players near the location where the ball have disappeared, and chases the players instead of trying to estimate the ball location. This mode transition is done on the basis of the number of meaningful pixels in the accumulation image. For each player who is suspected (marked) to have the ball, searching for the ball is done in a pre-determined area with the player position as the center. When a player comes close enough to the marked, it also becomes enlisted. After a detection of the re-appearance of the ball by counting the meaningful pixels, the proposed algorithm resumes ball tracking.

Temporary occlusion by a player causes the ball to appear to be stopped and kicked by him even though he never touches it. Excluding those spurious cuts from the ball trajectory completes the event detection by identifying real kickers and receivers.

Sequential Monte-Carlo method is explained in Section 2. Section 3 deals with pre-image processing and player tracking. The method of ball tracking is discussed in 4. Section 5 provides experimental results and finally Section 6 concludes this paper.

2 Sequential Monte-Carlo Algorithm

Particle filtering or sequential Monte-Carlo (SMC) algorithm estimates the posterior distribution $p(x_t|z_t)$ sequentially, where x_t is the state and z_t is the mea-

surement at time t, given a sequential dynamic equation with Gauss-Markov process.

The posterior is represented by random particles or samples from the posterior distribution. When it is not possible to sample directly from the posterior distribution, a proposal distribution q of known random sampler can be adopted to compute the posterior, and in this case the posterior at time t is represented by the set of pairs of particle s and its weight w updated sequentially:

$$w_t = w_{t-1} \frac{p(x_t|z_t)p(x_t|x_{t-1})}{q(x_t|x_{0:t-1}, z_{1:t})} \tag{1}$$

After computation of w_t's for the particles generated from q and normalization $\sum_1^N w_t^i = 1$, where N is the number of particles, the set of particles comes to represent the posterior distribution. Particles have the same weight $1/N$ after re-sampling based on the weights or the posterior distribution.

Taking the proposal distribution as $q = p(x_t|z_{t-1})$ results in $w_t = w_{t-1}p(z_t|x_t)$, saying that the posterior can be estimated by evaluating the likelihoods at each time using the particles generated from the prediction process of system dynamics. Incorporated with resampling, the weight update equation can be further reduced to $w_t = p(x_t|z_t)$, where weight normalization is implied afterwards. This is the method of *condensation* algorithm [13,14].

To solve the problem at hand by the condensation algorithm, one needs design appropriately the likelihood model $p(z|x)$ and state dynamic model $p(x|x_{t-1})$. In this paper, the random proposal particles are not generated from $p(x|x_{t-1})$ in the ball tracking, but from a novel proposal distribution taking account of the accumulated measurements. Therefore, we use Equation 1 for updating the weights for the posterior density.

3 Pre-image Processing and Player Tracking

The field part of original soccer image, I_k^{ogn} at frame k is subtracted to yield field-free image I_k^{sub} using histogram as in Figure 1. In I_k^{sub}, the pixels of field parts are marked as black. Via CCL (connected component labeling) I_k^{ccl} is obtained. Size filtering deletes colored blobs that have either bigger or smaller enough size not to be considered as those of people.

Player tracking is done in the way of [12]. For the image I_k^{sub} of k th frame, state estimates of players are done by the particle filter assigned respectively.

4 Ball Tracking

The basic idea in ball tracking is that the image consists of the players, ball and static background. So we may get I_t^{ball}, the image of ball only at the frame number t if we remove the portions of the background and players from the image.

While player tracking is done at every single frame, ball tracking is batch processed at every m-th frame, where the interval of ball tracking is to produce

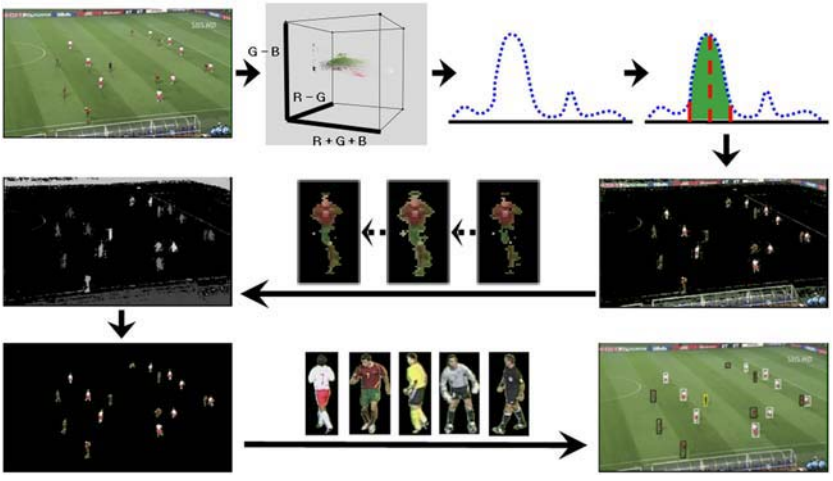


Fig. 1. Image processing

a long enough accumulated area of the ball blobs. Examples of the accumulation are shown in Figure 2 and in our experiments the ball tracking interval m was 20 frames. If the blobs of players are deleted completely from the background-free image, we can get an accumulation image of I^{ball}s that is supposed to contain white pixels only from the ball area. However, notice that it contains noise pixels, too, due to incomplete background removal and players' blob detection. One could see that the ball has been in *visible* mode through the sequence since there are white accumulated areas (the linear structure in the accumulation image). The discontinuity means that the ball has been *invisible* during a period due to some reasons such as occlusion and overlapping. During the visible mode, we use a first order dynamic model for the ball motion perturbed by Gaussian random noise η:

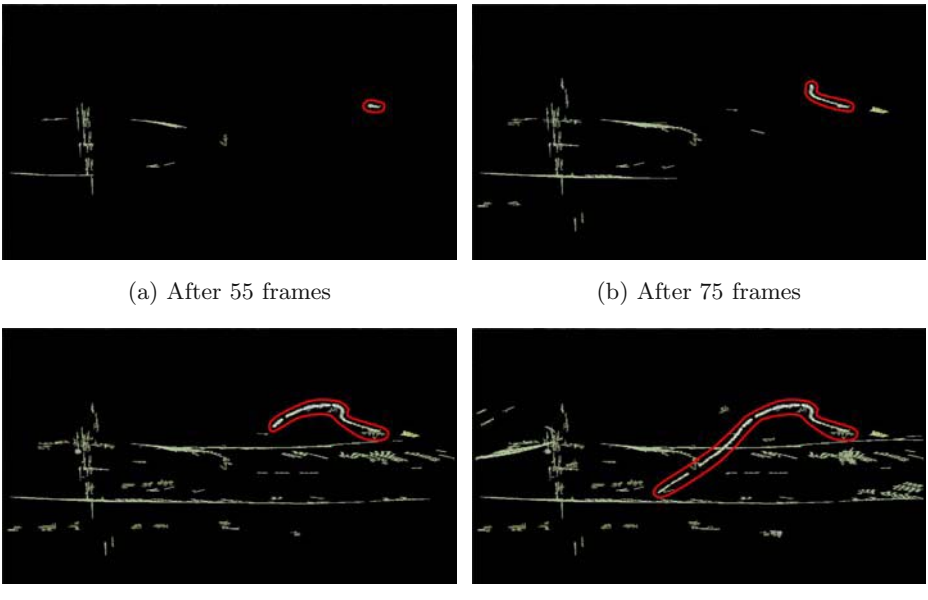
$$\mathbf{x_t} = \mathbf{2x_{t-1}} - \mathbf{x_{t-2}} + \eta, \tag{2}$$

where $\mathbf{x} = (\mathbf{x}, \mathbf{y})$ is the location of the ball. The shape of the ball is modelled simply to be 3×3 rectangular. We measure the color values on the pixels in the 3×3 rectangle whose center is given by $\mathbf{x}$ - the state of the ball motion. Hence, our observation model for a ball particle is defined to be:

$$p(\mathbf{z_t}|\mathbf{x_t}) = \prod_{\mathbf{i}} \prod_{\mathbf{c}} \exp\left(-\frac{(\mathbf{c_i} - \mu_\mathbf{c})^\mathbf{2}}{\sigma_\mathbf{c}^\mathbf{2}}\right), \tag{3}$$

where i denotes a pixel location i in the 3×3 rectangle, c_i the value in RGB color space at the pixel location, and μ_c and σ_c the mean and standard deviation calculated based on the pixel values around the ball area in a few video frames. Particles for the tracking is generated in the image region detected as the ball area after removing the players' blob and the background.

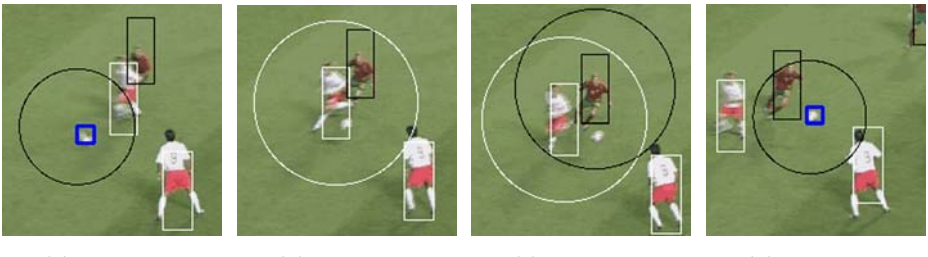
Those pixels are designed to have equal probability and hence a uniform random sampler is utilized. The likelihood is evaluated using Equation 1, and the ball location is given by the weighted average of the particles. When the ball is in the mode of *invisible*, we stop tracking the ball. In this case, the ball is assumed to be possessed by players near the place where the ball has become invisible. As shown in Figure 3, for each of the players who are suspected to have the ball, ball searching is done in the circled area with the player position as the center. Any player who comes close enough to the suspects also becomes enlisted. After the ball reappears and is detected through the accumulation, that is, one end of another ball blob trajectory (e.g. Figure 2) is found, the proposed



(a) After 55 frames (b) After 75 frames

(c) After 115 frames (d) After 135 frames

Fig. 2. Accumulation images for the ball blobs



(a) frame 353 (b) frame 357 (c) frame 359 (d) frame 362

Fig. 3. Sub-images of some frames of interest

algorithm resumes normal ball tracking as in the early part of this section. In order to determine the ball tracking mode, we observe the number of pixels of the ball area in the accumulation image. At the frame number t ($t \neq 0$ and $(k-1)m \leq t < km$ for a natural number k), this value is given as the sum:

$$S_t = \sum_{j \in \{t-1,t,t+1\}} \sum_{l \in W_t} C_j(\mathbf{x_l}), \tag{4}$$

where $\mathbf{x_l}$ denotes an l-th pixel location in the search window W_t whose center is given by the estimated ball position at the frame number t, and C_j is an indication function:

$$C_j(\mathbf{x}) = \begin{cases} 0 & \text{if the color at } I_j^{ball}(\mathbf{x}) \text{ is black} \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

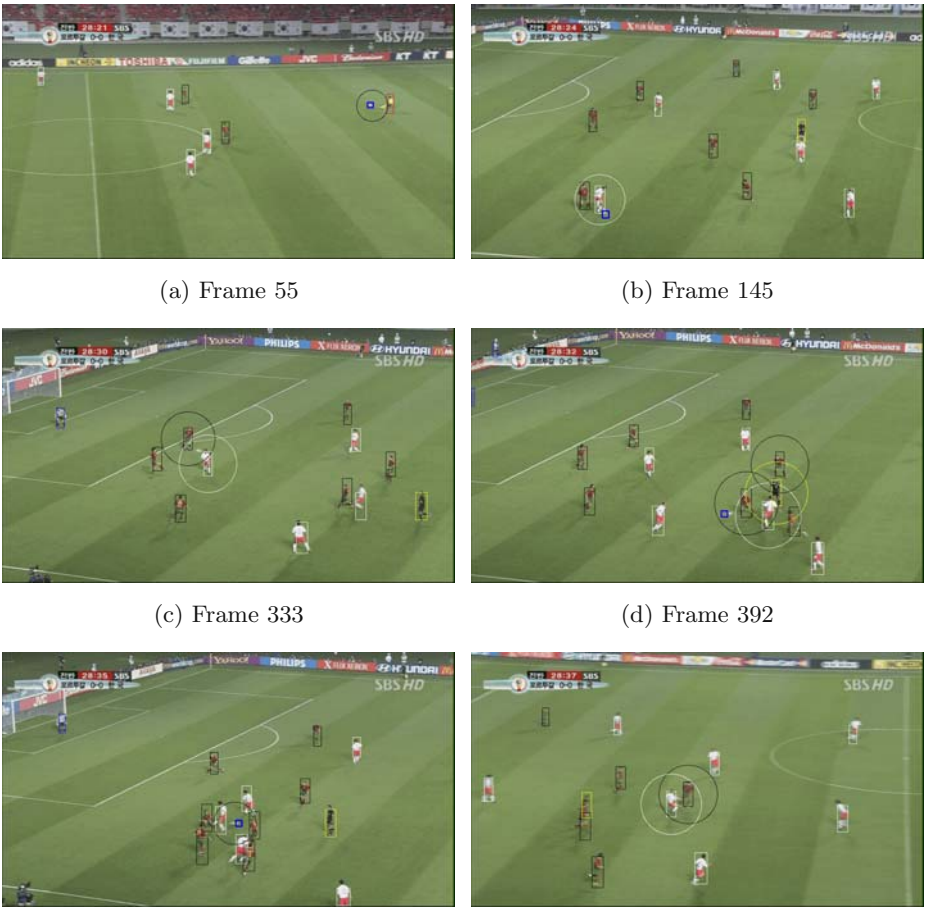
Note that we incorporate the three consecutive image measurements in Equation 4 for a robust computation. Mode change is done simply by thresholding. When S_t is smaller than a threshold Th then the tracking mode changes to *invisible*, and as we explained before, the players are kept traced until our tracker finds the re-appearance of the ball pixels, that is, $S_t \geq Th$. At the most frames of *ivisible* mode S_t is zero and over 50 for the *visible*. When $S_t < 15$ in the real experiment, the mode changed to *invisible* and nearby players were traced to find the initiation of the ball blobs.

5 Experiments

Experiments were carried out on a video sequence of 600 images whose size is 960×540 pixels. Figure 4 shows some frames of the results of which the detail is contained in accompanying video clip. The rectangle around each player is colored to show his class: ordinary players of each team, goal keeper of each team, and referee. A black circle around the ball means that the ball is not occupied by any player and thus the tracking mode is *visible*, and a colored circle shows the search area whose center is given by the location of the player, who is marked as a candidate having the ball. Notice that the color of the circle and the rectangle of the player are the same. The interval m was 20 and the threshold, Th for the mode transition was set to 15.

6 Conclusion

The algorithm presented in this paper have focused on an effective way of tracking the ball in a soccer match video broadcast on TV. The result of multiple player tracking was made use of in order to obtain a robust measurement for the ball tracking. By removing the blobs of players, we could obtain an accumulation image of the ball blobs. This accumulation image provided us not only a proposal density for the particle filtering but also a clue to deciding whether the ball was visible or invisible in the video frames. Basically, the ball tracking was done by



(a) Frame 55 (b) Frame 145

(c) Frame 333 (d) Frame 392

(e) Frame 465 (f) Frame 541

Fig. 4. Examples of result images

particle filtering. However, the performance was highly improved by two ingredients: first, taking the accumulation image as the proposal density, and second, mode change by counting the meaningful ball pixels. When the ball was invisible, we pursued every nearby players until the ball pixel came out again. Since the ball pixels were accumulated in time, the tracking algorithm showed in the real experiment a very robust ball tracking results, that was not shown by other studies. By excluding cuts in ball trajectory blob due to temporary occlusion, pairs of kicker and receiver are decided to extract events in the sequence.

Acknowledgment. This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

References

1. Intille, S., Bobick, A.: Closed-world tracking. In: Proc. Int. Conf. on Computer Vision. (1995)
2. Yow, D., Yeo, B., Yeung, M., Liu, B.: Analysis and presentation of soccer highlights from digital video. In: Proc. Asian Conf. on Computer Vision. (1995)
3. Seo, Y., Choi, S., Kim, H., Hong, K.: Where are the ball and players? soccer game analysis with color-based tracking and image mosaick. In: Proc. Int. Conf. on Image Analysis and Processing, Florence, Italy. (1997)
4. Iwase, S., Saito, H.: Tracking soccer player using multiple views. In: IAPR Workshop on Machine Vision Applications. (2002)
5. Ok, H., Seo, Y., Hong, K.: Multiple soccer players tracking by condensation with occlusion alarm probability. In: Int. Workshop on Statistically Motivated Vision Processing, in conjunction with ECCV 2002, Copenhagen, Denmark. (2002)
6. Yoon, H., Bae, Y., Yang, Y.: A soccer image mosaicking and analysis method using line and advertisement board detection. ETRI Journal **24** (2002)
7. Utsumi, O., Miura, K., IDE, I., Sakai, S., Tanaka, H.: An object detection method for describing soccer games from video. In: IEEE International Conference on Multimedia and Expo (ICME). (2002)
8. Kang, J., Cohen, I., Medioni, G.: Soccer player tracking across uncalibrated camera streams. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS). (2003)
9. Yamada, A., Shirai, Y., Miura, J.: Tracking players and a ball in video image sequence and estimating camera parameters for 3d interpretation of soccer games. In: Proc. International Conference on Pattern Recognition. (2002)
10. Yu, X., Xu, C., Leong, H., Tian, Q., Tang, Q., Wan, K.: Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In: ACM MM03, Berkeley. (2003) 11–20
11. Tong, X.F., Lu, H.Q., Liu, Q.S.: An effective and fast soccer ball detection and tracking method. In: ICPR (4). (2004) 795–798
12. Choi, K., Seo, Y.: Probabilistic tracking of the soccer ball. In: Int. Workshop on Statistical Methods in Video Processing, in conjunction with ECCV 2004, Prague, Czech Republic. (2004)
13. Kitagawa, G.: Monte-carlo filter and smoother for non-gaussian nonlinear state space model. Journal of Computational and Graphical Statistics (1996)
14. Blake, A., Isard, M.: Active Contours. Springer-Verlag (1997)
15. Liu, J., Chen, R.: Sequential Monte Carlo methods for dynamic systems. (1998)
16. Doucet, A., Godsill, S., Andrieu, C.: On sequential monte-carlo sampling methods for bayesian filtering. (2000)
17. Doucet, A., Freitas, N.D., Gordon, N., eds.: Sequential Monte Carlo Methods in Practice. Springer-Verlag (2001)

Automatic Roadway Geometry Measurement Algorithm Using Video Images

Yichang (James) Tsai[1], Jianping Wu[1], Yiching Wu[1], and Zhaohua Wang[2]

1 Geographic Information System Center, Georgia Institute of Technology,
276 fifth street, Atlanta, GA 30332, United States
{James.tsai@ce, jw248@mail, yw23@mail}.gatech.edu
[2] School of Civil and Environmental Engineering, Georgia Institute of Technology,
790 Atlantic Drive, Georgia Tech, Atlanta, GA 30332, United States
zw12@ce.gatech.edu

Abstract. The Georgia Department of Transportation (GDOT) collects and maintains an inventory of all public roads within the state. The inventory covers more than 118,000 centerline miles (188,800 km) of roads in 159 counties and over 512 municipalities. The transportation road inventory includes more than 52 items, including roadway geometry, surface type, shoulder type, speed limit signs, etc. Traditional roadway geometric properties, including number of lanes, travel lane, and shoulder widths, are measured in the field. Roadway geometric property measurement is one of the most important and, yet, the most time-consuming and riskiest component of the roadway data inventory. For the past two years, GDOT has sponsored Georgia Tech to develop a GPS/GIS-based road inventory system that re-engineers the existing paper-pencil operations. Georgia Tech has extended the research to develop video image pattern recognition algorithms and a prototype application aimed at automating the roadway geometry measurement to enhance the roadway inventory operations. A highly reliable and effective image extraction algorithm using local thresholding, predictive edge extraction, and geometric optics was developed and is presented in this paper. Preliminary results show it can effectively extract roadway features. A large-scale, experimental study on accuracy and the productivity improvement is under way.

1 Introduction

Collecting roadway infrastructure data, including pavement geometric properties (number of lanes, travel lane and shoulder widths), signs (stop signs, speed limit signs, etc.) at each location (milepost and x, y coordinates) is crucial for state Departments of Transportation (DOTs) in the US for submitting Highway Performance Monitoring System (HPMS) data annually. It is also important for transportation system planning, design, maintenance, and operations, including safety studies and traffic modeling. However, collecting road inventory data is a costly, time-consuming, and sometimes dangerous. The data quality and data update frequency cannot meet the growing transportation needs. Currently, many state DOTs have used video logging as a means for collecting roadway infrastructure data. The roadway video logging images are collected and then displayed on a computer screen, frame by frame, and the various

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 669 – 678, 2005.

roadway features, such as number of lanes, pavement width, and type and location of signs, are manually measured/extracted. Although the field data collection method, in terms of data quality and safety, has been improved using this approach, the process of extracting roadway information from the images remains time-consuming and tedious. The current practice of manually viewing and extracting roadway data from millions of video images taken at a typical 25 ft. interval would take an enormous amount of manpower and cost million of dollars to extract roadway data for each state DOT in the US. Thus, there is a need to automate this process. Our research is motivated by this need. We have developed video image sign recognition ([1], [2]) and roadway geometry features to address this need. This paper focuses on introducing algorithms to automate roadway geometry measurement using video images.

In the past, the edge mark recognition is usually developed for the purpose of automatic navigation of vehicles or intelligent drive assistance. The biggest challenge to lane detection methods are bad environmental conditions, such as low contrast, occluded lane markings, vehicles, puddle, ruts, shadows on a road surface, and so on. Weather and lighting are also major factors in bad conditions. There are various research groups that have tackled lane edge detection. The lane detection [3] is performed by finding possible edge features extracted from search windows set in a road image. The search windows handle a part of a whole image, but noise edge features are mistaken of lane markings when noise edge features increase in bad conditions. Some systems try to improve robustness by using a global road shape constraint and by using pixel data extracted from all over the image. Kluge [4] uses a least median of square estimation to determine the curvature and orientation of a road. Lane markings are individually found by a horizontal intensity profile. Though the least median of square estimation is robust for noisy data, real-time response is not assured because of its repetitive computations. Pomerleau [5] accomplishes an adaptively high-speed matching procedure to determine at lane's curvature and lateral offset. It has the advantage of handling any type of cue corresponding to a lane edge on the road surface. Though this algorithm, it reduces computation cost for rapid response and results in low precision due to neglected parameters. Kluge [6] and Kreucher et al. [7] lane detection maximizes the function that estimates the possibility of how well particular lane model parameters match the pixel data in a road image. They used a Metropolis algorithm to maximize the function, which is a random walk like stochastic optimization using an annealing process. Though the Metropolis algorithm achieves robust estimation, it also sometimes fails to find a global maximum. Kreucher [8] presented another algorithm based on the frequency domain features to extract edge information that seems to work well under conditions in which the spatial domain algorithm failed. Takahashi [9] presented an edge detection algorithm based on a global voting process in which the patterns with the greater number of votes are chosen as the best approximations of the left and right lane markings. Using multiple cues and particle filtering, Apostolof [10] provided another novel approach to lane detection. Gern [11] made use of optical flow to successfully extract the lane edges under adverse weather conditions. Xu [12] used a linear lane model to do the lane tracking. Bertozzi [13] introduced a more comprehensive set of various algorithms for lane edge detection. For the transform between image and world coordinates, Bucher [14] presented a method mapping image coordinates with world coordinates that we adopted into our algorithm.

In order to achieve our goal of automatically measuring pavement geometric properties (i.e. number of lanes, widths of travel lane and shoulder widths) using video images, we have developed an algorithm using color segmentation, edge mark extraction, and roadway geometry measurement. We have organized this paper as follows. The need for developing such an image pattern recognition algorithm and application is first introduced along with a brief literature review. An algorithm is proposed and presented in the subsequent section. The strength and limitations of the proposed algorithm are, also, discussed in the final section.

2 Proposed Algorithm

With the need discussed in the previous section, we have developed an image pattern recognition algorithm using local thresholding and predictive edge extraction. We have used pavement markings to identify the travel lane width, and we have used the contrast between grass and pavement to identify the shoulder width. The algorithm consists of three parts: image color segmentation, edge mark extraction, and roadway geometry measurement. The objective of image segmentation is to separate yellow and white edge marks and non-pavement backgrounds, such as grass, woods, or buildings, from the pavement surface background to facilitate the edge mark extraction. Edge mark extraction extracts all the white and yellow edge marks (as regions of interest) and combines broken edge marks if necessary and puts them in order from left to right. In this stage, false edges are also eliminated. The last stage is geometric information collection, which makes use of the result of extraction and calculates the travel lane width, travel lane number, shoulder width, etc. The following are the detailed presentations of the algorithms that we used.

2.1 Image Color Segmentation

In the segmentation, we adopt two types of approaches to work complementally. The first segmentation method makes use of the intensity and color information to segment the white and yellow edge marks from the pavement background. The second method makes use of the gradient to segment the boundary between pavement and background. In some highways, edges are not marked either by yellow or white, in which case the most efficient way to determine edges is by means of gradient because the gradient in the boundary area is usually much higher than internal area. The segmentation of non-white-non-yellow edges and shoulder border requires an algorithm totally different from previous algorithms. The Sobel Gradient to detect non-white and non-yellow edges and shoulder edges is used. Due to the content space limitation, this section presents only the first segmentation method.

Pavement surface is, generally, gray, which is dominated by blue or green but with very little color saturation. The yellow edge marks are dominated by two color components, green and red, and are usually brighter than the pavement surface. The white edge marks have about the same level of intensities in all three colors (red, green, and blue) and are significantly brighter than the pavement surface. A grass or woods background is dominated by green and generally has a much lesser intensity than pavement surfaces. The greatest challenge, however, is that the pavement surface has a widely variable intensity, not only because of the non-uniformity of pavement

surface (existence of black and white spots, also known as pepper and salt noise), but also because of the variation of brightness along the pavement surface due to different reflectivity levels from different angles into the camera. What is more, usually in every travel lane, the middle area is always darker than the area close to a lane border, primarily because the black asphalt at the top of the pavement changes to a lighter color with the increased abrasion by vehicle wheels, and the two sides of the lane always receive far more wheel wear than the middle of the lane. The variation in brightness between the lane middle areas and the side areas can reach up to 30% of average intensity, in which case a global threshold to segment pavement surfaces usually does not work very well. To compound the problem, the white edge lines tend to lose their brightness and become less distinguishable from the pavement surface because of aberration and corrosion. Sometimes, using cameras with different responses (filters) can result in different tints for the same scene, and a weakly blue pavement surface could change. Figure 1 shows the flow chart for the color segmentation. The threshold will have to be dynamic to succeed in separating the background from the edge lines. To make a reliable pavement surface segmentation, we use the following steps:

1. Calculate the average r_{mean}, g_{mean}, b_{mean} and standard deviation r_{std}, g_{std}, and b_{std} of each of the three-color components in two image locations. One is located in the bottom (size of m*n) of the image, and the other is located in the middle area (size of m'*n'). Here m, n, m', n' are parameters that can be adjusted to result in the best result. In our experiment, m is half of image width, n is one tenth of image height, m' is a quarter of image width, and n' is one tenth of image height. Interpolation of these calculated averages and standard deviations reveal the approximate average and standard deviation distribution in the full range of pavement surface of the image.
2. Classify each pixel based on the following criteria:
    a. A pixel is regarded as a pavement pixel if
    $r-b<k*|r_{mean}-b_{mean}|$ and $g-b< k*|r_{mean}-b_{mean}|$ and $|r-r_{mean}|<r_{std}$
    or
    $| r - r_{mean} |< \alpha * r_{std}$ and $| g - g_{mean} |< \alpha * g_{std}$ and $| b - b_{mean} |< \alpha * b_{std}$
    b. A pixel is regarded as a potential yellow edge pixel if it does not satisfy criterion (a) and if
    $r > \lambda * b$ and $g > \lambda * b$ and $r > r_{mean}$
    c. A pixel is regarded as a white one if it does not satisfy criteria (a) and (b), and it satisfies:
    $r > r_{mean} + \rho \cdot r_{std}$ and $g > g_{mean} + \rho \cdot g_{std}$ and $b > b_{mean} + \rho \cdot b_{std}$
    d. A pixel that does not meet any of the above criteria is called non-pavement background pixels.

In the above, *r*, *g*, and *b* are the red, green, and blue intensities for a pixel, and r_{mean}, b_{mean}, *and* g_{mean} are the interpolated red average, blue average, and green average, and r_{std}, g_{std}, and b_{std} are the interpolated standard deviations of the red,

green, and blue averages on the pavement surface. α, λ and ρ are constants that are related with the camera and pavement conditions. The average and standard deviation are used to remove the pavement background. To get the pavement width for the situation where there are no white or yellow edge marks that separate the pavement from non-pavement, the Sobel Gradient is adopted.

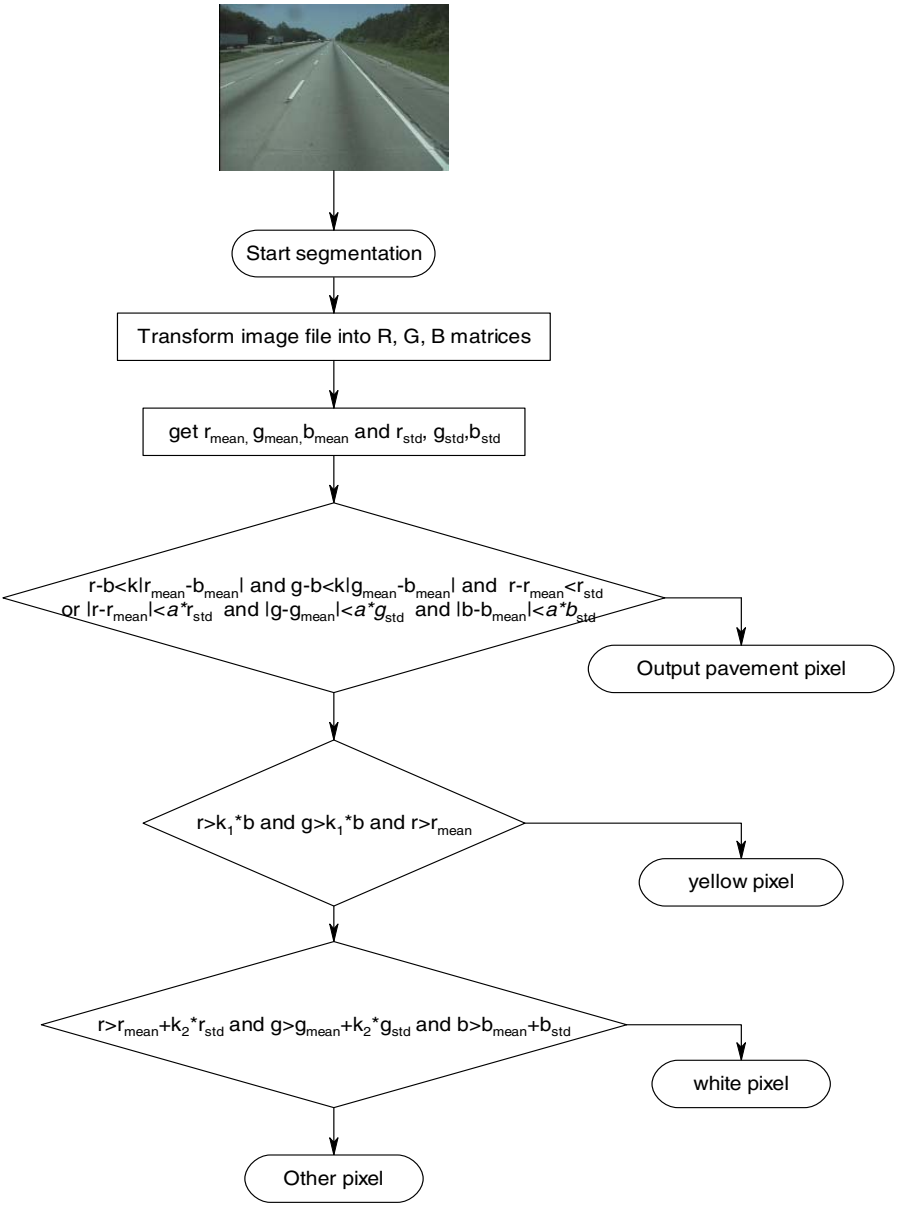


Fig. 1. Flow chart of color segmentation

2.2 Edge Extraction and Validation

Figure 2 shows the flow chart for edge extraction. Edge extraction makes use of a depth-first search algorithm to search every white or yellow region of interest (ROI) and to mark every pixel with a certain ROI ID number. Then, we do the linear fit for each ROI and calculate the slope *a*, intercept *b*, and the standard deviation STD of the pixels along the line. For a group of points $(x_1, y_1),(x_2, y_2),....,(x_n, y_n)$ in a certain ROI, we assume that $x = ay + b$ passes through the middle of this ROI. Then Equation (1) is used to calculate *a and b*:

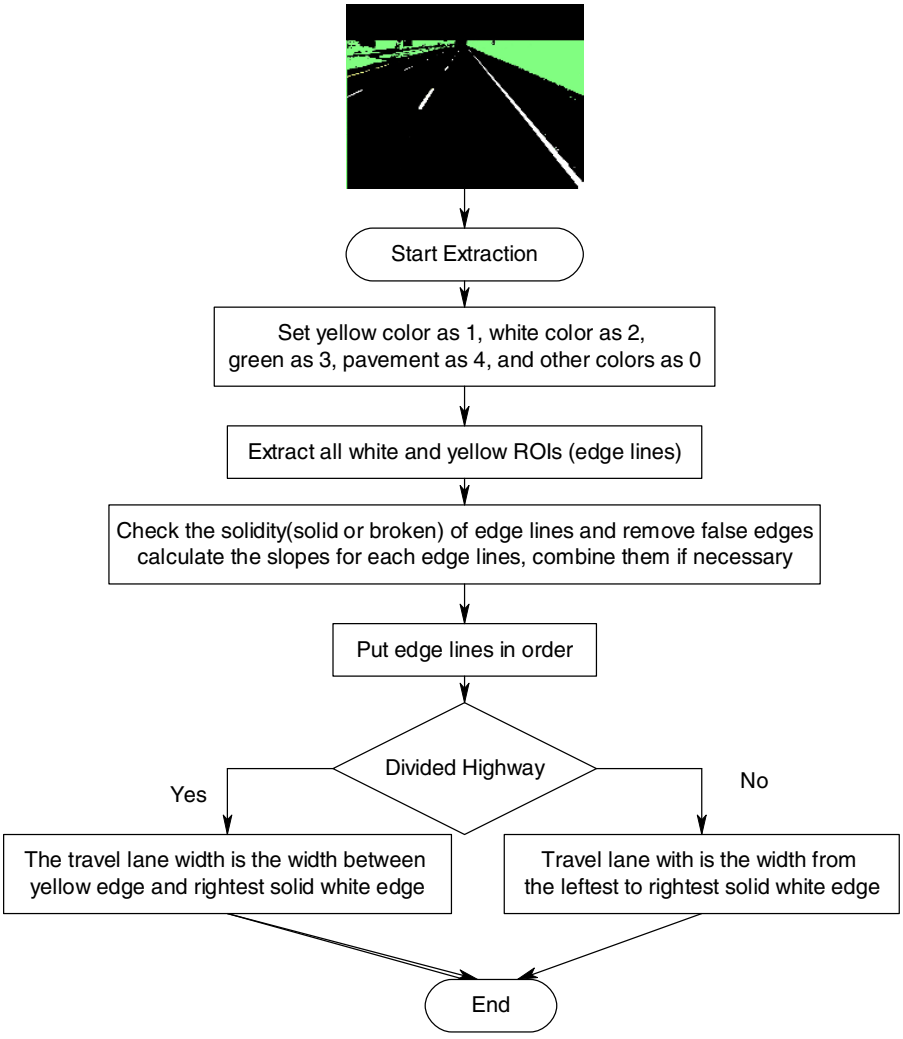


Fig. 2. Flow chart for edge mark extraction

$$\begin{cases} a = \dfrac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2} \\ b = \dfrac{n\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} x_i y_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2} \end{cases} \quad (1)$$

To indicate how close to the line of all points in the ROI are, we use standard deviation (STD) as a benchmark. An ROI that is narrow and very close to a straight line has a smaller STD than an ROI that is wide or curved. The formula to calculate STD is:

$$STD = \sqrt{\frac{\sum x_i^2 - a\sum x_i y_i - b\sum x_i}{n}} \quad (2)$$

After getting the slope, intercept, and STD for all ROIs, they are put in ascendant order according to their slopes. If an ROI is surrounded by a rectangular box described by (x_1, y_1) and (x_2, y_2), and has the total pixel number of N_{pixel}, then they are validated one by one first using the criteria:

a. A valid ROI has the top of the ROI below the vanishing point.
b. A valid ROI has the height of $(y_2-y_1+1)>20$
c. A valid ROI satisfies: STD<0.3*height
d. A valid ROI satisfies: $(y_2-y_1+1)*STD<N_{pixels}$
e. A valid ROI's drawn line will pass through more same color pixels than other color pixels.
f. A valid yellow ROI edge's slope is always negative.
g. A valid ROI edge always has a distance from the camera's vanishing point smaller than a certain threshold (i.e. 50).

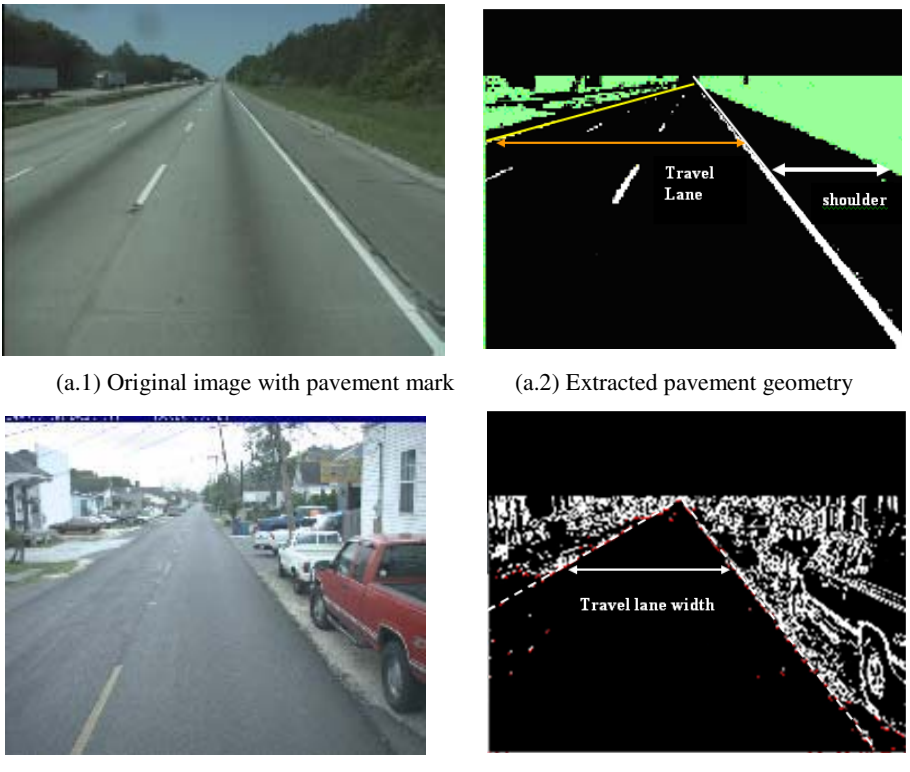
After the above validation process, there may still be redundant edge lines. The following processes are used to remove redundant edges and both lines are then determined:

a. First, set those edges whose lengths are longer than a certain threshold as solid yellow or solid white lines.
b. If there exist both solid yellow and solid white edge lines and if the highway is divided, then we have already got the traveling lane edges, and we can directly go to shoulder edge detection step.

c. If at least one side does not have a solid edge, then further analysis is needed to determine whether there exists a partly blocked solid edge. If all edges are found to be lane-separating marks that don't determine the pavement edges, we need to use the Sobel Gradient to detect real pavement edges. If it is a blocked solid edge, then this side is finished. The Sobel Gradient is used to detect the edge, which will be described below, in which case no shoulder detection process is needed.

2.3 Roadway Geometry Measurement

After all edge marks, including those non-white-non-yellow edges, have been extracted, geometric optics are used to compute the pavement geometric information, including pavement width, number of lanes, shoulder width, and possibly the intersection information. To calculate the travel lane width and shoulder width, the pavement area is first identified. In the pavement area, if both left-most edge mark and right-most edge mark are identified, then the travel lane width is calculated by means of geometric optics. Before that, the necessary calibration is needed to determine how many inches a



(a.1) Original image with pavement mark (a.2) Extracted pavement geometry

(b.1) Original image without pavement mark (b.2) Extracted pavement geometry

Fig. 3. Original images and the extracted roadway geometry

pixel stands for in each row. If we assume the vanishing point position in an image is at $P(x_v, y_v)$, then the distance that per pixel stands for is:

$$d = \frac{k}{y_v - y} \tag{4}$$

For a divided highway, travel lane width has a left border of yellow edge line and right border of solid white edge lines. The shoulders are usually situated left from the yellow edge and right of the most-right solid white edge. The widths of the shoulders are usually calculated from the edge line to the borders of pavement surface. For an undivided highway, the travel lane is usually calculated from the leftmost solid white edge line to the rightmost solid edge lines, and shoulder widths are calculated from the pavement boundary to two solid white lines. For the case where no pavement edge lines exist, the derived edges from gradients by means of the Hough Transform are adopted. Here, shoulders are ignored. The number of lanes in the travel lanes is derived by dividing travel lane width by 11 feet and rounding to the nearest integer.

Figure 3 shows the examples that roadway geometry, including travel lane and shoulder lane widths and number of lanes, can be automatically extracted using video images with the developed algorithm. Figures 3.a and 3.b show that the developed algorithm can handle the roadway with and without pavement marks.

3 Summary

Traditional roadway geometric properties, including number of lanes, travel lane and shoulder widths, are measured in the field. Roadway geometric property measurement is one of the most important and, yet, the most time-consuming and riskiest component of the roadway data inventory. This paper presents an automatic roadway geometry feature extraction algorithm using a highly reliable and effective image extraction algorithm using local thresholding, predictive edge extraction, and geometric optics. This methodology makes segmentation very accurate for color images, and it dynamically adjusts the segmentation criteria and is able to deal with widely different lighting conditions. This algorithm combines statistics-based color segmentation, depth-first search edge extraction, geometric optic measurement, and edge detection based on gradient and Hough Transform. It utilizes localized color segmentation to minimize the impact from non-uniform pavement intensity over the image view field. It also uses recursive iteration to get the accurate edge line slope and intercept and result in more accurate measurement of lane width and shoulder width measurement. The sequential images are processed and used to improve the reliability of automatic pavement measurement performance. The preliminary study shows that the developed algorithm can automatically extract roadway geometry effectively and will improve current roadway inventory dramatically.

A large-scale experimental study using video images collected by state DOT is under way. The accuracy of the extracted results will be evaluated, and the improved productivity will be measured. One of the most important features of pavement geometrical property is the curvature. We will implement a reliable highway horizontal

curvature algorithm to measure horizontal curvature. It is also suggested to incorporate and post-process GPS data to assist in roadway geometry validation and calibration.

References

1. Tsai, Y. and Wu, J.: Shape and Texture-based 1-D Image Processing Algorithm for Real-time Stop Sign Road Inventory Data Collection, Journal of Intelligent Transportation System, Vol. 7, pp. 213-234.
2. Wu, J. and Tsai, Y. (2005), Speed Limit Extraction and Recognition Algorithm Using Locally Adaptive Thresholding and Depth-First-Search, Photogrammetric Engineering and Remote Sensing (PE&RS) Journal. (in press)
3. Behringer, R.: Road Recognition from Multifocal Vision, *Proceedings of IEEE Intelligent Vehicle 1994*, Paris, France, pp.302-307, 1994.
4. Kluge, K.: Extracting Road Curvature and Orientation From Image Edge Points Without Perceptual Grouping Into Features, *Proceedings of IEEE Intelligent Vehicles 1994*, Paris, France, pp**.** 109- 114
5. Pomerleau, D.: RALPH:Rapidly Adapting Lateral Position Handler, *Proceedings of IEEE Intelligent Vehicle Symposium*, Detroit, MI, USA, September, 1995, pp.506-511.
6. Kluge, K.: A Deformable-template approach to lane detection, *Proceedings of IEEE Intelligent Vehicle Symposium*, Detroit, MI, USA, September, 1995, pp.54-59.
7. Kreucher, C., Lakshmanan, S., and Kluge, K.:, A Driver Warning System based on the LOIS Lane Detection Algorithm, *Proceedings of IEEE International Conference on Intelligent Vehicles*, Stuttgart, Octobre 28-30 1998, pp.17-22.
8. Kreucher, C. and Lakshmanan, S.: LANA: A Lane Extraction Algorithm that Uses Frequency Domain Features. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, VOL. 15, NO. 2, APRIL 1999, pp343-350.
9. Takahashi, A., Ninomiya, Y., Ohta M., and Tange KA Robust Lane Detection using Real-time Voting Processor, *Proceedings of the IEEE Intelligent Transportation Systems*, Tokyo, Japan, October 5-8, (1999), pp. 76-79.
10. Apostoloff, N., and Zelinsky, A.: Robust Vision based Lane Tracking using Multiple Cues and Particle Filtering, *Proceedings of the IEEE Intelligent Vehicles Symposium 2003*, Columbus, Ohio, USA, June, 9-11, 2003 pp. 558-563.
11. Gern, A., Moebus, R., Franke, U.:. Vision-based Lane Recognition under Adverse Weather Conditions Using Optical Flow , *Proceedings of the IEEE Intelligent Vehicles Symposium 2002*, Versailles, France, Jun 17-21, 2002 pp. 652-657.
12. Xu, Y., Wang, R., Li, B., Ji, S. A Vision Navigation Algorithm Based on Linear Lane Model. *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, Dearborn (MI), USA October 3-5, pp. 240-245. (2000).
13. Bertozzi, M. Broggi, A., Cellario, M., Fascioli, A., Lombardi, P., and Porta, M.:(2002). Artificial Vision in Road Vehicles, *Proceedings of the IEEE - Special issue on Technology and Tools for Visual Perception*, 90(7):1258-1271, July 2002
14. Bucher, T.: Measurement of Distance and Height in Images based on easy attainable Calibration Parameters, *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, Dearborn(MI), USA. October 3-5, 2000, pp. 314-319.

An Improved Algorithm for Anchor Shot Detection

M. De Santo[1], G. Percannella[1], C. Sansone[2], and M. Vento[1]

[1] Dipartimento di Ingegneria dell'Informazione e di Ingegneria Elettrica,
Università di Salerno - Via P.te Don Melillo, 1 I-84084, Fisciano (SA), Italy
{desanto, pergen, mvento}@unisa.it
[2] Dipartimento di Informatica e Sistemistica, Università di Napoli "Federico II",
Via Claudio, 21 I-80125 Napoli (Italy)
carlosan@unina.it

Abstract. Segmentation of news videos into stories is among key issues for achieving efficient treatment of news-based digital libraries. Indeed, anchor shot detection is a fundamental step for segmenting news into stories.

In this paper we present an improved algorithm for anchor shot detection. It is based on a graph theoretical clustering method and exploits the idea of the *cluster lifetime* for improving the performance. Moreover, a method for automatically fixing all the thresholds required by the original version of the algorithm is also proposed, so making it fully unsupervised.

The proposed algorithm has been tested on a database significantly wider than those typically used in the field, demonstrating its advantages with respect to the original version.

1 Introduction

Among all the different sources of video material nowadays available, news videos received great attention by the scientific community. This is mainly due to the fact that broadcasters are interested in building large digital databases of their resources, so as to allow the reuse of the archived material for other TV programs. Such a reuse can be efficiently performed only after a suitable indexing procedure of the existing materials, where an important step towards effective indexing is the segmentation of a news video into *stories*.

At a first stage, this implies the partition of the video into *shots*, i.e. sequences of frames, obtained by detecting transitions that are typically associated to camera changes. Once shots have been individuated, they can be classified on the basis of their content. Two different classes are typically considered: *anchor* shots and *news-report* shots. Successively, the given news video can be segmented into stories. This is obtained by linking each anchor shot with all successive news report shots until another anchor shot, or the end of the news video, occurs. Using this model for the stories, anchor shot detection algorithms can be employed as basic step for segmenting news videos.

In the literature, most of the approaches to the anchor shot detection problem exploit the video source information by using a model matching strategy [1]. For each shot, a distinctive frame, called *key-frame*, is extracted. Then, it is matched against a

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 679 – 686, 2005.

set of predefined models of an anchor shot frame in order to classify it. This approach is strongly dependent on the model of the specific video program. This is a severe limitation, since it is difficult to construct all the possible models for the different news videos and the style of a particular news program can change over the time.

Other authors use a face detection approach to identify anchor shots [2]. However, face detection in video is generally too time-consuming for practical application. Furthermore, a shot where a reporter (not an anchorman!) is present can be erroneously recognized as an anchorperson shot by the face detection module.

In order to overcome the above reported limitations, some authors [3,4,5] propose methods that are unsupervised and do not require the explicit definition of an anchor shot model. Among them, one of the most effective algorithms is that proposed by Gao and Tang in [3]. Here, anchor shots are identified by using an algorithm based on graph-theoretical cluster analysis. It automatically groups similar key-frames into clusters, on the basis of their color histograms. The key-frames composing a cluster are classified as potential anchorperson frames if the cluster size is greater or equal to 2. Then, a spatial difference metric (SDM) is used to refine the shot classification; in fact, in some situations, the key-frames in a cluster may have similar histograms but different content. If a cluster has an average SDM value higher than a suitable threshold, it is removed from the anchorperson frame list. As pointed out by the authors, however, this approach fails when identical or very similar news-report shots appear in different stories of the same news program.

A more recent version of the algorithm has been proposed by Gao et *al.* in [6]. Here, the idea presented in [5] is adopted for reducing the number of false alarms (i.e., falsely detected anchor shots). In particular, starting from the key-frames of the cluster with maximal size, a template is generated and used to remove clusters having key-frames significantly different from it. The template is obtained in an unsupervised way, so preserving one of the most distinctive features of the algorithm. However, this approach cannot be used when there are two anchorpersons in the news video. In theses cases, in fact, there are at least three different anchorperson models, while the template obtained by following the approach proposed in [5] is unique for each news video.

In this paper we propose an improved version of the algorithm presented in [6] that uses the idea of the *cluster lifetime* (borrowed from [4]) to deal with the false alarms provided by the algorithm proposed by Gao and Tang. In particular, the temporal interval (*lifetime*) that includes all the occurrences of key-frames belonging to a cluster is evaluated. Since anchorperson shots repeatedly occur along the whole video, the clusters having a lifetime smaller than a suitably fixed threshold are removed from the anchorperson list.

Moreover, as noted by the authors in [6], the algorithm proposed by Gao and Tang needs that some parameters are specified in advance. In this paper we also propose a method for automatically fixing the two thresholds required by the original version of the algorithm and experimentally prove that the threshold on the lifetime value can be fixed in a straightforward way.

In order to test the modified algorithm in a significant way, we built-up a database that is considerably bigger than those typically used in the field [3]. Namely, we used a news video database consisting of about 17 hours with 673 anchor shots and 8922 news report shots. Since in this database there are also news videos presented by two anchorpersons, in our tests we consider as benchmark only the first version of the algorithms proposed by Gao, i.e. the one presented in [3].

The organization of the paper is as follows: in section 2, the original version of the algorithm is recalled and the proposed modifications are presented. In section 3, the database used is reported together with the tests carried out in order to assess the performance of the proposed algorithm. Finally, in section 4, some conclusions are drawn.

2 The Algorithm

The authors in [3] propose to classify video shots by using an algorithm based on graph-theoretical cluster (GTC) analysis. More in details, they propose an anchor shot detection scheme composed of four steps: short shot filtering, key-frame extraction, GTC analysis and post-processing.

In general, an anchorperson shot should last for more than 2 sec, since this shot should involve at least one sentence pronounced by the reporter. Therefore, if a shot lasts less than 2 sec it is considered as a news report shot. Otherwise, it is further analyzed through later steps.

The second step is the key-frame extraction: the authors propose that the middle frame is taken as the key-frame. These key-frames are the input to the GTC analysis module. It considers them as vertices in a feature space and then builds the minimum spanning tree (MST) [8] on these vertices. For constructing the MST it is necessary to associate a weight to each edge connecting two vertices. So, a distance between two key-frames needs to be defined. This distance is the weight associated to the edge that connects the two vertices representing the key-frames in the feature space. Each key-frame is divided into 16 regions of the same size; the histograms of corresponding regions in the two key-frames are compared and the eight regions with the largest histogram differences are discarded to reduce the effects of object motion and noise. The distance between these two key-frames is then defined as the sum of the histogram differences of the remaining regions. By using this distance, the MST can be constructed. Successively, by removing all the edges in the tree with weights greater than a threshold γ, a *forest* containing a certain number of subtrees (*clusters*) is obtained. In this way, the GTC method automatically groups similar vertices (key-frames) into clusters. The key-frames composing a cluster are classified as potential anchorperson frames if the size of the cluster is greater or equal to 2.

Starting from this set of potential anchorperson frames, the last step of the proposed detection scheme operates a further filtering. In fact, in some situations, the key-frames in a cluster may have similar histograms but different content. To detect this situation, a spatial difference metric (*SDM*) between two key-frames *KF1* and *KF2* in a cluster is proposed:

$$SDM = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| I_{KF1}(i,j) - I_{KF2}(i,j) \right| \tag{1}$$

where $I_{KF1}(i,j)$ and $I_{KF2}(i,j)$ denote the intensity of a pixel at location (i,j) in the frames $KF1$ and $KF2$ respectively, and the frame size is M x N.

If a cluster has an average *SDM* higher than a threshold λ, the whole cluster is removed from the anchorperson frame list. By using the *SDM* filtering, the final anchorperson shot classification is obtained.

It is worth noticing that the above anchorperson shot detection scheme requires to specify in advance two thresholds: γ for the GTC algorithm and λ for the post-processing step, respectively.

2.1 The Modified Version

As anticipated in the introduction, the main contribution of this paper is twofold: on one side we try to improve the performance of the algorithm proposed by Gao and Tang by removing some false alarms it produces, while on the other side we propose a method for automatically fixing the two thresholds γ and λ required by the original version of the algorithm, so making it completely unsupervised.

In order to remove the false alarms generated by the algorithm proposed by Gao and Tang, we add a further filter after the post-processing stage. This filter is based on the idea of the *cluster lifetime* (borrowed from [4]). It is defined as the temporal interval that includes all the occurrences of key-frames belonging to a cluster. The rationale of this kind of filter lies in the observation that anchorperson shots repeatedly occur along the whole video. Therefore, clusters having a lifetime smaller than a suitably fixed threshold δ should not belong to the anchor class and then can be removed from the anchorperson list.

This filter is very effecting in dealing with the very frequent situation of a person interviewed: in this case there are at least two shots (Fig. 1.*a* and 1.*b*) whose key-frames are very similar each other, separated by one or more shots in which there is the presence of a reporter (or the anchorman himself). The shots represented by similar key-frames are grouped together into a cluster by the GTC and the average SDM value of the cluster is very low: as a result the original algorithm erroneously attributes the shots to the anchor shot class. However, the proposed filter recognizes these shots as a news report since their cluster lifetime is very small, as it can be noted by considering the difference between the indexes of their key-frames shown in Fig. 1.

(a) (b)

Fig. 1. Key-frames extracted from two news videos of our database that are erroneously detected as anchor shots by the original algorithm, but correctly classified by the modified version proposed here

It is worth noting that the filter based on the lifetime also uses a threshold; however, as already noted in [4], the threshold on the lifetime value can be fixed in a straightforward way on the basis of the length of the specific news program (more details about this point will be given in the experimental section).

The second contribution provided in this paper consists in a method for automatically computing the two thresholds γ and λ required by the original version of the algorithm.

As regards γ, our proposal is to determine its value by reformulating the problem as the one of partitioning the whole set of edges into two clusters, according to their weights. The cluster of the edges of the MST with small weights will contain edges to be preserved, while the edges belonging to the other cluster will be removed from the MST. In order to solve this problem we employ the Fuzzy C-Means (FCM) clustering algorithm.

FCM is a clustering technique based on the minimization of the following objective function:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m}\left(x_i - c_j\right)^2, \quad 1 \le m < \infty \tag{2}$$

where m is any real number greater than 1, x_i is the i-th measured data (in our case the weight of the i-th edge of the MST), c_j is the center of the cluster, u_{ij} is the degree of membership of x_i to the cluster j, C is the number of clusters (in our case $C = 2$) and N is the number of objects to be clustered. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{x_i - c_j}{x_i - c_k}\right)^{\frac{2}{m-1}}} \qquad \text{and} \qquad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{3}$$

This iteration will stop when:

$$\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon \tag{4}$$

where ε is a termination criterion between 0 and 1, whereas k are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m. At the end of the procedure, each edge x_i has been assigned to the cluster r such that:

$$r = \arg\max_{j} u_{ij} \tag{5}$$

At this point, all the edges of the MST are separated into two clusters. Then, we remove from the MST all the edges belonging to the cluster s whose center exhibits the largest value, i.e.:

$$s = \arg\max_{j} c_j \tag{6}$$

As regards the threshold λ, we consider the K clusters obtained by using the GTC analysis in increasing order with respect to the value of the *SDM*. Let *SDM*(0) be the smallest value and *SDM*(K-1) the largest values. Then, λ is calculated as the average value of the first N SDM values, where the parameter N is simply related to the number of anchorpersons of each news program. In fact, we use $N = 4$ if the news video has only one anchorperson, otherwise $N = 6$. This choice is justified by the fact

that in the editions presented by two anchorpersons there are typically two additional anchor shot models with respect to the editions with a single anchorperson.

In order to strengthen the method against possible SDM outliers, we prefer to discard *SDM*(0) during the computation of the average value. So, the value of λ is calculated as follows:

$$\lambda = \frac{1}{N-1}\sum_{i=1}^{N-1} SDM(i) \tag{7}$$

However, this procedure cannot be effectively applied when the number of clusters resulting from the GTC analysis is equal or even less than three. In a news video, in fact, there are in general at least three different anchor shots models. In this case we do not activate the SDM filtering; the cluster lifetime will discard false clusters, if any.

3 Experimental Results

Some efforts have been spent in the recent past by other researchers in building video databases for benchmarking purposes; in particular in [7] a database was built in order to characterize the performance of shot change detection algorithm. This database, however, is not adequate for our aims, since it is made up not only of news videos but also of sport events and sitcom videos, and the duration of news videos is only 20 minutes. To reproduce as much as possible the variability of the phenomenon under study, different news video editions of a single broadcaster should be considered, as well as news videos of different broadcasters. For this reason, the database used in this paper (about 17 hours) is composed by 28 news videos from the main Italian public network (namely, RAI 1) and 17 videos from the main Italian private network (namely, CANALE 5). Particular care was taken in order to include in the database the main news editions from these broadcasters. As it can be easily noted from Table 1, the size of our database is large; this is more evident if it is compared with the database used by Gao and Tang in the paper [3]. The performance of the proposed algorithm was then assessed on the two TV-networks, separately. Note that all the editions of RAI 1 are presented by a single anchorperson, while the news videos of CANALE 5 are presented by two anchorpersons, even if some anchor shots of these news videos are characterized by the presence of a single anchorperson.

Table 1. Composition of the databases used in this paper and in [3]

Paper	Total length (hh:mm:ss)	Number of videos	Number of Broadcasters	Number of Anchor/News-report shots
This	16:54:09	45	2	673 / 8922
[3]	05:05:17	14	2	253 / 3654

When dealing with unbalanced data sets, like in this case, where the anchor shot samples outnumber news report shot samples, the performance is typically reported in terms of *Precision* and *Recall*. However, in order to compare the two algorithms a single figure of merit should be used. In particular, we used the parameter *F* defined in [9], which combines *Precision* and *Recall* as in the following:

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{8}$$

As pointed out in Sect. 2, the original algorithm is characterized by the two thresholds λ and γ. Then, different operating points can be obtained in a *Precision-Recall* plane. Differently, the modified version of the algorithm requires to select only the value of the threshold δ on the lifetime value, since the thresholds λ and γ are automatically calculated. As regards the original version of the algorithm, we decided to choose the values of the thresholds that maximize *F* over the whole set of videos. This has been done separately for each of the two TV-networks. Obviously, this is an overestimation of the real performance of the algorithm, since such maximization should be done on a different set of news videos. It is also worth noting that, as experimentally demonstrated in [10], the choice of the operating point is crucial for the algorithm, as its performance dramatically depends on the choice of the thresholds λ and γ. On the contrary, we experimentally verified that the value of δ can change in a wide range without affecting the performance of the modified algorithm. In particular, it depends only on the length of the news edition, and can be fixed to 2 m for all the news editions shorter than fifteen minutes and to a value equal to 4 m otherwise. Therefore, our algorithm can be really considered as fully unsupervised.

Table 2. The performance of the original algorithm by Gao and Tang and the modified algorithm proposed in this paper in terms of *Precision*, *Recall* and *F*

	RAI 1			**CANALE 5**		
	Recall	***Precision***	***F***	***Recall***	***Precision***	***F***
Original	0.854	0.947	0.898	0.812	0.681	0.741
Modified	0.923	0.994	0.957	0.852	0.930	0.889

Table 2 reports the performance of each algorithm for the two considered TV-networks. For the news videos of both the TV-networks there is a noteworthy improvement in terms of *F*. This represents a really valuable result if we consider that the algorithm in [3] is already characterized by a good performance.

Table 3. The performance of the original algorithm by Gao and Tang and the modified version using *i*) only the lifetime control, *ii*) only the automatic threshold computation and *iii*) both the proposed modifications

	RAI 1			**CANALE 5**		
	Recall	***Precision***	***F***	***Recall***	***Precision***	***F***
Original	0.854	0.947	0.898	0.812	0.681	0.741
Original + Lifetime	0.886	0.985	0.933	0.840	0.930	0.884
Original + Automatic Thresholds	0.928	0.923	0.926	0.852	0.761	0.804
Both Modifications	0.923	0.994	0.957	0.852	0.930	0.889

In order to better understand the contribute to the performance improvement given by the introduction of the cluster lifetime and the automatic threshold computation, we calculated the value of the *Precision*, *Recall* and *F* on the two datasets by

considering the original algorithm by Gao and Tang using only the lifetime control or only the automatic threshold computation. The obtained results are reported in Table 3. From the results reported in Table 3 it is evident that both the proposed modifications allow us to improve the performance with respect to the original algorithm. In particular, the lifetime control allows the proposed algorithm to significantly outperform the original algorithm of Gao and Tang in terms of *Precision*. On the other hand, the automatic selection of the thresholds permits a significant improvement in terms of *Recall*, especially for the RAI 1 videos.

4 Conclusions

In this paper an improved algorithm for anchor shot detection was presented. It was tested on a news video database consisting of about 17 hours, giving rise to a significant improvement with respect to its original version.

In order to further improve the *Recall* value obtained by the proposed algorithm, other information sources, such as the audio track of the news video, could be used. Moreover, other similarity measures between key-frames could be employed for dealing with errors due to the fact that some anchorperson models appear only once in a whole news program. All these matters will be subjects of future investigations.

References

1. B. Furht, S.W. Smoliar, H. Zhang, Video and Image Processing in Multimedia Systems, Kluwer Publishers, Boston (MA), 1996.
2. Y. Avrithis, N. Tsapatsoulis, S. Kollias, “Broadcast news parsing using visual cues: A robust face detection approach”, Proc. of IEEE ICME, vol. 3, pp. 1469–1472, 2000.
3. X. Gao, X. Tang, “Unsupervised Video-Shot Segmentation and Model-Free Anchorperson Detection for News Video Story Parsing”, IEEE Trans. on Circuits and Systems for Video Technology, vol. 12, no. 9, pp. 765-776, 2002.
4. M. Bertini, A. Del Bimbo, P. Pala, “Content-based indexing and retrieval of TV News”, Pattern Recognition Letters, vol. 22, pp. 503-516, 2001.
5. A. Hanjalic, R.L. Lagendijk, J. Biemond, “Semi-Automatic News Analysis, Indexing, and Classification System Based on Topics Preselection”, Proc. of SPIE: Electronic Imaging: Storage and Retrieval of Image and Video Databases, San Jose (CA), 1999.
6. X. Gao, J. Li, B. Yang, “A Graph-Theoretical Clustering based Anchor Shot Detection for News Video Indexing”, Proc. of Int. Conf. on Comput. Intell. and Multimedia App., 2003.
7. U. Gargi, R. Kasturi, S.H. Strayer, “Performance Characterization of Video-Shot-Change Detection Methods”, IEEE Trans. on Circuits and Systems for Video Technology, vol. 10, no. 1, pp. 1-13, 2000.
8. R. Balakrishnan, K. Ranganathan, A Textbook of Graph Theory. New York: Springer-Verlag, 1999.
9. L. Chaisorn, T.-S. Chua, C.-H. Lee, “A Multi-Modal Approach to Story Segmentation for News Video”, World Wide Web, vol. 6, pp. 187–208, 2003.
10. M. De Santo, G. Percannella, C. Sansone, M. Vento, “A Comparison of Unsupervised Shot Classification Algorithms for News Video Segmentation”, Lecture Notes in Computer Science vol. 3138, Springer, Berlin, pp. 233-241, 2004.

Probabilistic Detection and Tracking of Faces in Video

G. Boccignone[1], V. Caggiano[2], G. Di Fiore[3], and A. Marcelli[1]

[1] Natural Computation Lab – DIIIE, Università di Salerno,
via Ponte Don Melillo, 1 Fisciano (SA), Italy
{boccig, amarcelli}@unisa.it
[2] Dipartimento di Informatica e Sistemistica,
Università di Napoli, via Claudio 21, Napoli, Italy
vcaggian@unina.it
[3] Co.Ri.Tel. Lab, via Ponte Don Melillo, 1 Fisciano (SA), Italy
gdifiore@coritel.it

Abstract. In this note it is discussed how face detection and tracking in video can be achieved relying on a *detection-tracking loop*. Such integrated approach is appealing with respect either to robustness and computational efficiency.

1 Introduction

Face detection and tracking can be performed either according to a frame based approach, in which faces are detected in each frame, or according to a detection and tracking approach, where the faces are detected in the first frame and tracked through the video sequence (for a review, refer to [1]). Clearly, in the first case temporal information is not exploited, while in the second case a loss of information may occur (e.g., new faces entering the scene). Here, we propose a system which relies on a *detection-tracking loop*, where the tracking step go in a stand-by state when face displacement from one frame to another is not significant. The tracking is accomplished through a color and shape based Particle Filtering and along tracking, exchange of information occurs between the detection and filtering modules. The architecture of the system is outlined in Fig. 1.

2 Interleaved Detection and Tracking

We assume that the video is a sequence $(f_1, f_2, \cdots, f_t, \cdots)$ represented in the YC_rC_b color space. In our system, at the frame t of the video, the face detection module detects one or more new faces, each face being characterized by a state $x_t = \{x, y, wX, wY, \theta\}$ where x, y specify the location of a rectangular box (or ellipse) surrounding the face, wX and wY the length of the half axes and θ is the rotation angle between the vertical side of bounding box and vertical axes. The objective of tracking a face is to estimate the state x_t given all the measurements $Z_t = \{z_1, \cdots, z_t\}$ up to that moment, or equivalently to construct the posterior

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 687–694, 2005.

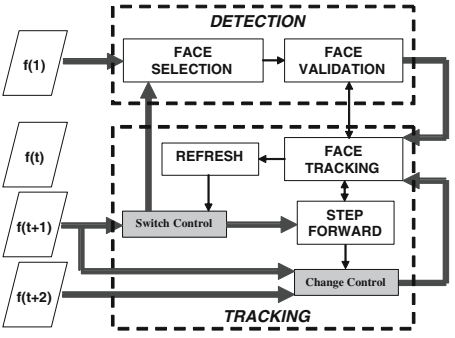


Fig. 1. The proposed system at a glance.

probability density function (pdf) $p(x_t|Z_t, \mathcal{F})$, $\mathcal{F}$ being the class of faces. The theoretically optimal solution is provided by the recursive Bayesian filter which solves the problem in two step. The *prediction* step uses the dynamic equation and the already computed pdf of the state at time $t-1$, $p(x_{t-1}|Z_{t-1}, \mathcal{F})$, to derive the prior pdf of the current state, $p(x_t|Z_{t-1}, \mathcal{F})$. Then, the *update* step employs the likelihood function $p(z_t|x_t, \mathcal{F})$ of the current measurement to compute the posterior pdf $p(x_t|z_t, \mathcal{F})$. Formally:

$$p(x_t|Z_t, \mathcal{F}) \propto p(z_t|x_t, \mathcal{F}) \int p(x_t|x_{t-1}) p(x_{t-1}|Z_{t-1}, \mathcal{F}) dx_{t-1}. \tag{1}$$

We assume that the likelihood of observing a face, is the joint likelihood of observing face specific color and shape features, which can be factorized as:

$$p(z_t|x_t, \mathcal{F}) = p(z_t^{color}|x_t, \mathcal{F}) p(z_t^{shape}|x_t, \mathcal{F}). \tag{2}$$

In particular $p(z_t^{color}|x_t, \mathcal{F})$ is the likelihood of an histogram based color observation, while $p(z_t^{shape}|x_t, \mathcal{F})$ is the likelihood computed by the detection module with respect to a template $t(\cdot)$ representing a prior shape model of class $\mathcal{F}$. The system (see Fig. 1), at the beginning of the video sequence ($t = 1$) detects one ore more faces to track, each represented through a state vector x_1. For each face, an independent tracker is initialized. When, at frame f_t the pdf $p(x_t|Z_t, \mathcal{F})$ has been determined, before deriving the prediction at frame f_{t+1} a *refresh* step (see Fig. 1) controls the possible switching to detection, based on the number of frames elapsed from the last detection step, so as to detect new events (faces). New events are assumed as arrivals over discrete, no-overlapping temporal intervals accounted for by a Poisson process. If the detection module is not activated, a *step forward* module checks, by fast matching, wether within frame f_{t+1} the state of the tracked face has not changed ($x_{t+1} \approx x_t$): if that is the case, the current state is determined as $x_{t+1} = x_t$, thus avoiding the filtering step, and next frame f_{t+2} is considered; on the contrary, Bayesian filtering is performed.

Independent tracking of each face is motivated by the choice of Particle Filtering (PF) to implement the Bayesian filter, so as to achieve a good trade-off between quality of prediction and number of particles employed.

3 Face Detection

The face detection is accomplished in two steps: candidate face selection and candidate validation.

Face selection is organized in two processing streams: the first exploits a skin model and uses color information, the second performs eye detection by operating on the luminance channel. *Skin Detection* is achieved by deriving a preliminary skin map [2]. Then, the map is filtered in order to eliminate regions with non homogeneous hue. *Eye detection* is performed by taking into account circular symmetries and grey level variations. The center of regions exhibiting circular symmetries are derived via the Discrete Symmetry Transform (DST [3]). For a fixed radius r, a thresholding is performed, and candidate eye points are marked if $DST(x, y) > t$ where $t = \mu_D + 3\sigma_D$, and correspondingly a symmetry map is obtained. Then, by taking into account grey level variations as in [4] an eye analogue map is derived, which is eventually combined with the symmetry map trough an AND operation, obtaining a global eye map.

The list of candidate faces is obtained by combining skin and eye maps, through geometrical conditions followed by an AND operation; then, the face bounding region, denoted $\mathcal{R}_F$, is computed. Faces already detected are masked to avoid multiple detection of the same faces.

Face validation is accomplished by taking into account the joint likelihood of a candidate face, with respect to skin, texture and shape features, which is factorized as follows:

$$p(z_t^{skin}, z_t^{tex}, z_t^{shape}|x_t, \mathcal{F}) = p(z_t^{skin}|x_t, \mathcal{F})p(z_t^{tex}|x_t, \mathcal{F})p(z_t^{shape}|x_t, \mathcal{F}). \quad (3)$$

In order to compute $p(z_t^{skin}|x_t, \mathcal{F})$, the skin occupancy ratio in candidate face box is computed as $r_{skin} = \frac{n_{skin}}{|A|}$, where n_{skin} is the number of points in the skin map occurring in the face box, and $|A|$ the area of the box. Then,

$$p(z_t^{skin}|x_t, \mathcal{F}) = K_1 \cdot (1 - e^{-\beta' r_{skin}}). \quad (4)$$

To measure $p(z_t^{tex}|x_t, \mathcal{F})$, the textural similarity in the two cheek regions (areas below eyes and at the side of nose, referring to the eye map) is used as in [5], calculated as a function of the grey level variance V_Y, namely $R_{tex} = \frac{|V_Y^{left} - V_Y^{right}|}{V_Y^{left} + V_Y^{right}}$. Thus:

$$p(z_t^{tex}|x_t, \mathcal{F}) = K_2 \cdot (1 - \frac{K_3}{1 + e^{-\beta'' R_{tex}}}), \quad (5)$$

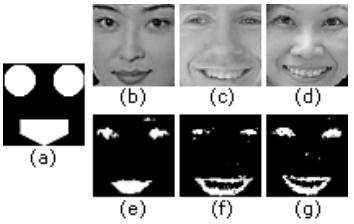


Fig. 2. Templates used for computing $p(z_t^{shape}|x_t, \mathcal{F})$ (a) Reference shape template $t(\cdot)$; (b), (c) and (d) candidate faces; (e), (f) and (g) eyes and mouth maps fused in a single shape map

For what concerns the shape likelihood, for each candidate face we directly locate eyes and mouth based on their feature maps derived from chromatic information, by using the method suggested in [6], and eventually determine the likelihood with respect to a reference binary template $t(\cdot)$ with support $\mathcal{R}_T$ (cfr. Fig. 2). Formally:

$$p(z_t^{shape}|x_t, \mathcal{F}) = p(z_t^{leye}, z_t^{reye}|x_t, \mathcal{F}) p(z_t^{mouth}|x_t, \mathcal{F}) \tag{6}$$

where, $p(z_t^{leye}, z_t^{reye}|x_t, \mathcal{F})$ and $p(z_t^{mouth}|x_t, \mathcal{F})$ denote the joint probability of observing the left and right eyes and the probability of observing a mouth with reference to the template $t(\cdot)$, respectively. $p(z_t^{leye}, z_t^{reye}|x_t, \mathcal{F})$ is obtained as:

$$p(z_t^{leye}, z_t^{reye}|x_t, \mathcal{F}) = \frac{K_4}{\sqrt{2\pi}\sigma_{eye}} e^{-\frac{(d_l+d_r+d_A)^2}{2\sigma_{eye}^2}} \tag{7}$$

where $d_l = \sum_{\mathbf{r}\in A_l} |e(\mathbf{r}) - t(\mathbf{r})|$, $d_r = \sum_{\mathbf{r}\in A_r} |e(\mathbf{r}) - t(\mathbf{r})|$, and $A_l, A_r \subseteq \mathcal{R}_T$ are the regions of left and right eyes in the template t, respectively, and d_A is the difference between the areas covered by each eye. The binary map $e(\cdot)$ is based on the observation that high C_b and low C_r values are found around the eyes [6], and computed as $e(\mathbf{r}) = \frac{1}{3}\{(C_b(\mathbf{r}))^2 + (C_r(\mathbf{r}))^2 + \frac{C_b(\mathbf{r})}{C_r(\mathbf{r})}\}$, followed by thresholding. Similarly, mouth likelihood is obtained as:

$$p(z_t^{mouth}|x_t, \mathcal{F}) = \frac{K_5}{\sqrt{2\pi}\sigma_{mouth}} e^{-\frac{(d_m)^2}{2\sigma_{mouth}^2}} \tag{8}$$

with $d_m = \sum_{\mathbf{r}\in A_m} |m(\mathbf{r}) - t(\mathbf{r})|$, where $A_m \subseteq \mathcal{R}_T$ and $m(\cdot)$ is the binary mouth map, computed under the assumption that the color of mouth regions contains stronger red component and weaker blue component than other facial regions [6]; thus, $m(\mathbf{r}) = C_r(\mathbf{r})^2 \cdot (C_r(\mathbf{r})^2 - 0.95 \cdot \frac{C_r(\mathbf{r})}{C_b(\mathbf{r})})^2$, followed by thresholding. Once $p(z_t|x_t, \mathcal{F})$ has been computed for each candidate face, the face is validated if $p(z_t|x_t, \mathcal{F}) > T$. K_1, K_2, K_3, K_4 and K_5 are normalizing factors that have been experimentally determined.

4 Face Tracking

Color and shape based tracking of detected faces is accomplished via Particle Filtering [7]. The main idea of PF relies upon approximating the probability distribution by means of a set of weighted samples $S = \{(s^{(n)},\ \pi^{(n)})\}$, $n = 1\ \ldots\ N$. Every sample s represents the current object status to which is associated a discrete sampled probability π, where $\sum_{n=1}^{N} \pi^{(n)} = 1$. The goal is to compute by sampling the a posterior probability $p(s_t|Z_t, \mathcal{F})$ in place of $p(x_t|Z_t, \mathcal{F})$. That is, filtering is performed by rewriting Eq. 1 as

$$p(s_t|Z_t, \mathcal{F}) \propto p(z_t|s_t, \mathcal{F}) \int p(s_t|s_{t-1}) p(s_{t-1}|Z_{t-1}, \mathcal{F}) ds_{t-1}, \tag{9}$$

and the same holds for Eq. 2:

$$p(z_t|s_t, \mathcal{F}) = p(z_t^{color}|s_t, \mathcal{F}) p(z_t^{shape}|s_t, \mathcal{F}). \tag{10}$$

In our case, $s_t = \{x,\ y,\ wX,\ wY,\ \theta\}$ and $\pi_t^{(n)} = p(z_t|x_t = s_t^{(n)}, \mathcal{F})$.

As with regards to S_t, after having selected N samples from the set S_{t-1} with probability $\pi_{t-1}^{(n)}$, prediction $p(s_t|s_{t-1})$ is obtained by propagating each sample by a first order dynamical model, $\widetilde{s}_t = \mu_s + (s_{t-1} - \mu_s) \cdot A + \sigma_s \cdot w_t$, where A describes a region $\mathcal{R}_F$ moving with constant velocity $(\dot{x}, \dot{y})$, μ_s is the mean status and σ_s is the variance of the noise factor w_t. Then, data observation is accomplished and the likelihood $p(z_t|s_t, \mathcal{F})$ evaluated.

For what concerns the shape likelihood in Eq. 10 it can be computed through Eq. 6, while the color likelihood $p(z_t^{color}|s_t, \mathcal{F})$, it is calculated as described in the following.

Each hypothetical face (particle) is specified by its state vector $s^{(n)}$. A target weighted histogram h_T, obtained from the detected face and the candidate weighted histogram $h_{s^{(n)}}$ in the region $\mathcal{R}_F$ specified by $s^{(n)}$ are calculated. In order to favor samples whose color distributions are similar to the target model, the color likelihood is specified as:

$$p(z_t^{color}|s_t, \mathcal{F}) = \frac{1}{\sqrt{2\pi}\sigma_{color}} e^{-\frac{d_B^2}{2\sigma_{color}^2}} \tag{11}$$

where d_B is the Bhattacharyya distance between the target histogram and the histogram of the hypotheses computed at point $\mathbf{r} = (x, y)$, $d_B(\mathbf{r}) = \sqrt{1 - \rho(\mathbf{r})}$, where $\rho(\mathbf{r}) = \rho[h_{s^{(n)}}(\mathbf{r}), h_T] = \sum_{u=1}^{m} \sqrt{h_{s^{(n)}}^u(\mathbf{r}) h_t^u}$, m being the number of histogram bins.

Weighted histograms are computed using an isotropic kernel, with a convex and monotonic decreasing profile, namely $k(x) = 1 - x^2$ if $x < 1$ and 0 otherwise, where x is the distance from center [8]. Color histograms $b(\mathbf{r})$ that associate to the pixel $\mathbf{r}$ the index of its bin u in the quantized feature space, are computed and the kernel based probability is $h^u(\mathbf{r}) = C \sum_{\mathbf{r}' \in \mathcal{R}} k\left(\frac{\|\mathbf{r} - \mathbf{r}'\|}{a}\right) \delta[b(\mathbf{r}') - u]$, where

$C = \sum_{\mathbf{r}' \in \mathcal{R}} k\left(\frac{\|\mathbf{r}-\mathbf{r}'\|}{a}\right)$, δ is the Kronecker delta function and the parameter $a = \max\{wX, wY\}$ is used to adapt the size of the region $\mathcal{R}$ from which the weighted histogram is computed; for the candidate face $\mathcal{R} = \mathcal{R}_F$.

Each particle is then weighted in terms of the observation with probability $\pi^{(n)}$. Eventually, the mean state of a object is estimated at each time step from $\mu_s = E[s] = \sum_{n=1}^{N} \pi^{(n)} s^{(n)}$, and this determines the position of the face. Note that, at time t, the sample selection from the sample set S_{t-1}, performed before the propagation step, is accomplished as described in [7].

5 Experimental Work

The *refresh* control (see Fig. 1) is based on the number of frames elapsed from the last detection, so as to detect new events (faces) s^{new}. These are assumed as arrivals over discrete, no-overlapping temporal intervals accounted for by a Poisson process, and the control is performed on the cumulative probability $P(s_t^{new}) = \frac{1}{2} \cdot \sum_{k=0}^{\lambda(t)} \frac{\mu^k}{k!} e^{-\mu}$, where $\lambda(t)$ measures the number of frames that have been tracked since last detection, k the frame counter, which is reset each time a new detection is performed, μ the average duration of the tracking step. Detection is performed when $P(s_t^{new}) > T_{new}$. The *step forward* control is mainly intended to gain computational efficiency. It is computed on the pair of frames (f_t, f_{t+1}) by fast change detection in the same region $\mathcal{R}_F$, i.e.

Fig. 3. The behavior of our system: when a face is detected it is tracked along the frames, even if it is partially occluded (first row). In case of totally occlusion, the tracking keeps sampling the same region until the faces reappears or a time-out elapses (second row). The appearance of another face is promptly detected and another tracker is activated (third row). In order to show the robustness of the proposed method, light changes have been introduced (as in the first row and third row), as well as both subject (first and second row) and camera movements (third row).

$\Delta = \sum_{\mathbf{r}\in\mathcal{R}_F} |f_t(\mathbf{r}) - f_{t+1}(\mathbf{r})|$. Hence the PF step is avoided if $\Delta < T_\Delta$, in order to take into account small variations (e.g., eyes blinking, lips movement), for which $S_{t+1} = S_t$.

The performance of the system has been assessed on a set of 29 videos whose duration ranges from 30 seconds to 2 minutes, and with a frame rate between 15 and 30 fps. The set is composed of both standard videos (such as *Akiyo*, *Mother & Daughter*, etc), and non standard ones, i.e. produced by us. 9 videos, 4 standard and 5 nonstandard, were used as training set for setting the parameters of the system, while the remaining (5 standard and 15 nonstandard) constituted the test set. Examples of the behavior of the system on a nonstandard video and on the standard video *Mother & Daughter* are summarized in Figs. 3 and 4, respectively. The system succeeds in detecting and tracking the faces in presence of changes in lighting, partial to complete face occlusion, camera movements.

Fig. 4. Results on the *Mother & Daughter* video.

Quantitatively, the face detection failed in 2 of the 20 videos, and all the failures were due to either too small faces or too low contrast between the face and the scene. In the first case, the DST failed in finding the eyes, while in the second one the skin detection was unable to find the faces or to separate the faces from the background. On the contrary no failures of the tracking were detected.

Eventually, Fig. 5 presents plots of the processing time due to face tracking on the *Akiyo* and *Mother & Daughter* videos. It can be noted that for negligible facial movements of the Akiyo speaker and of the daughter (top and center graphs), the step forward module, by inhibiting PF tracking increases the overall system performance. Time performance (msecs) has been measured using a Centrino 1600MHz CPU, under Windows XP operating system. Current software implementation is in Java language, without specific optimizations.

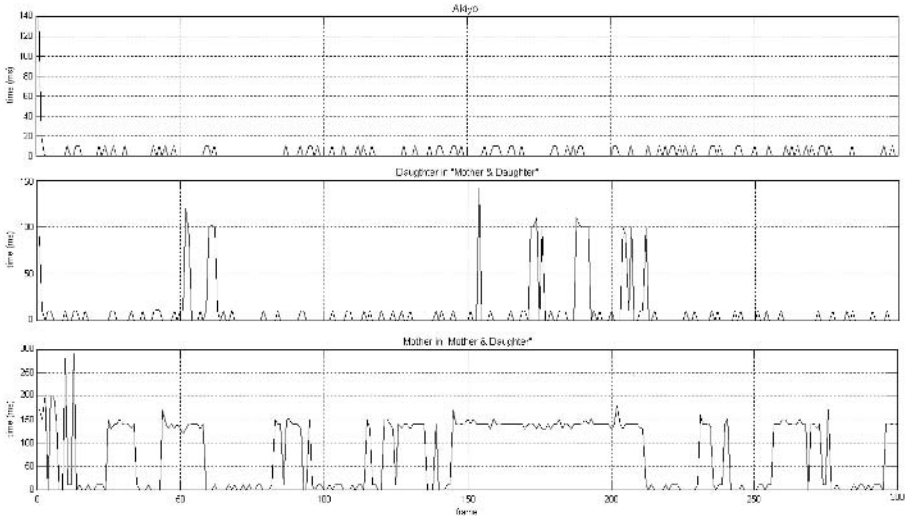


Fig. 5. Plots of the processing time (*msecs*) due to particle filter tracking of faces vs. frame number in *Akiyo* and *Mother & Daughter*. Top graph: *Akiyo*'s face PF tracking. Center: daughter's face. Bottom: mother's face.

References

1. R.C. Verma, C. Schmid and K. Mikolajczyk, "Face detection and tracking in a video by propagating detection probabilities", *IEEE Trans. on PAMI*, vol. 25, no. 10, 2003, pp. 1215-1227.
2. D. Chai and K.N Ngan, "Face Segmentation Using Skin-Color Map In Videophone Applications", *IEEE Trans. on CSVT*, vol. 9, no.4, 1999, pp. 551-564.
3. V. Di Gesu' and C. Valenti, "Symmetry operators in computer vision", *Vistas in astronomy*, vol. 40, no. 4, 1996, pp. 461-468.
4. Z.H. Zhou and J. Wu, "Efficient face candidate selector for face detection", *Patt. Rec.*, vol. 36, no. 5, 2003, pp. 1175-1186.
5. W. Huang and R. Mariani, "Face Detection and Precise Eyes Location", *Proc. 15th Int. Conf. on Patt. Rec.*, vol. 4, 3-7 Sept 2000, pp. 722-727.
6. R-L. Hsu, M. Abdel-Mottaleb. and A.K. Jain, "Face detection in color images", *IEEE Trans. on PAMI*, vol. 24, no. 5, 2002, pp. 696-706.
7. M. Isard, and A. Blake, "Condensation-Conditional Density Propagation for Visual Tracking", *Int. J. of Comp Vis.*, vol. 29, no. 1, 1998, pp. 5-28.
8. K. Nummiaro, E. Koller-Meier and L. Van Gool, "An adaptive color-based particle filter", *Im. and Vis. Comp.*, vol. 21, 2003, pp 99-110.

Removing Line Scratches in Digital Image Sequences by Fusion Techniques

Giuliano Laccetti[1], Lucia Maddalena[2], and Alfredo Petrosino[3]

[1] University of Naples ,"Federico II" Via Cintia, 80126 Naples, Italy
giuliano.laccetti@dma.unina.it
[2] Italian National Research Council, ICAR, Via P. Castellino 111, 80131 Naples, Italy
lucia.maddalena@na.icar.cnr.it
[3] University of Naples "Parthenope", Via A. De Gasperi 5, 80133 Naples, Italy
alfredo.petrosino@uniparthenope.it

Abstract. Many algorithms have been proposed in literature for digital film restoration; unfortunately, none of them ensures a perfect restoration whichever is the image sequence to be restored. Here, we propose an approach to digital scratch restoration based on image fusion techniques for combining relatively well settled distinct techniques. Qualitative results are deeply investigated for several real image sequences.

1 Introduction

With the recent advent of digital technologies, and the ever increasing need for speed and storage, the removal of occluded or missing parts in images and movies is a more and more widespread problem. The problem can occur in several multimedia applications, such as wireless communication and digital image sequence restoration. Several classes of defects can be distinguished that affect movies, and many algorithms have been proposed in literature for their restoration; unfortunately, none of them ensures a perfect restoration whichever is the image sequence to be restored.

A sufficiently general model of degraded video signal is the following for a pixel location $\mathbf{p} = (x, y)$:

$$I(\mathbf{p}, t) = (1 - b(\mathbf{p}, t))I^*(\mathbf{p}, t) + b(\mathbf{p}, t)c(\mathbf{p}, t) + \eta(\mathbf{p}, t) \ , \tag{1}$$

where $I(\mathbf{p}, t)$ is the corrupted signal at spatial position $\mathbf{p}$ in frame t, $b(\mathbf{p}, t) \in \{0, 1\}$ is a binary mask indicating the points belonging to missing parts of the degraded video, I^* is the ideal uncorrupted image. The (more or less constant) intensity values at the corrupted spatial locations are given by $c(\mathbf{p}, t)$. Though noise is not considered to be the dominant degrading factor in the defect domain, it is still included in (1) as the term $\eta(\mathbf{p}, t)$.

In the present paper, we focus on the class of scratch defects, intended as long and thin vertical scratches that affect several subsequent images of a sequence, due to the abrasion of the film by dust particles in the slippage mechanisms used for the development, projection and duplication of the film.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 695–702, 2005.

Commonly, scratch restoration is a two-step procedure. In the first step the scratches need to be detected, i.e., an estimate for the mask $b(\mathbf{p}, t)$ is made (*detection step*). In the second step the values of I^* inside the scratch, possibly starting from information about $c(\mathbf{p}, t)$, are estimated (*removal step*). As usual, we consider scratch reduction as a problem of detection and removal of missing data, i.e. we suppose that any information $c(\mathbf{p}, t)$ has been lost within the scratch.

Several scratch restoration methods are reported in literature (see for instance refs. [1] through [13]). As expected, advantages and disadvantages characterize each scratch detection and removal technique, and any of them could be said to win the competition. A way to deal with this kind of problems is to adopt *fusion techniques* (see for instance [14,15]): input images which provide alternative and complementary "views" and "characteristics" of a given area are "fused" into a single image. Fusion techniques should ensure that all the important visual information found in input images is transferred into the fused output image, without the introduction of artifact distortion effects. In this sense machine vision systems can be organized as a set of separated visual modules that act as virtual sensors. In this paper, the term *visual module* indicates an algorithm that extracts information of some specific and descriptable kind from a numerical image. For what concerns digital scratch restoration, a fusion technique may be applied both to the detection stage and to the removal stage: in both cases it takes into account already existing promising algorithms and suitably combines the obtained results in order to provide a restored sequence as similar as possible to the original uncorrupted sequence. As we show, the results produced by the proposed approach upon different damaged movies greatly enhance those produced by each considered approach. The experimental results reported in literature showed that the accuracy provided by the combination of an ensemble of visual modules can outperform the accuracy of the best single visual module.

The contents of this paper are as follows: Section 2 outlines the proposed compound algorithm for scratch restoration, based on scratch detection and removal modules and suitable detection and removal fusion strategies. Section 3 describes the qualitative results achieved by the proposed fusion approach tested on real video sequences. Conclusions are reported in Section 4.

2 Proposed Algorithm

The considered algorithm for scratch restoration in image sequences is based on an approach that takes into account already existing promising algorithms and suitably combines the obtained results in order to provide a restored sequence as similar as possible to the original uncorrupted sequence [16]. The basic idea of the *compound algorithm* consists, for each sequence frame, in:

1. applying a set of d existing scratch detection algorithms;
2. combining obtained scratch masks $B^j, j = 1, \ldots, d$, to produce the final scratch mask B^C;

3. applying a set of r existing scratch removal algorithms using scratch mask B^C;
4. combining obtained restored images $R^j, j = 1, \ldots, r$, to produce the final restored image R^C.

For the implementation of the compound algorithm we have considered as underlying restoration modules three detection algorithms presented in [7,8,9] and two removal algorithms presented in [3,11]. Moreover, for the combination of results we used fusion techniques which allow to exploit the performance of underlying algorithms while reducing their drawbacks [16], briefly described in the following. With such techniques input images, providing alternative and complementary views of a given region, are fused into a single image, in such a way that all the relevant content of input images is transferred to the fused output image, without introducing distortions.

2.1 Detection Fusion Strategies

The main goal of using more than one detection visual module is to make up for deficiencies in the individual modules, while retaining their features, thus achieving a better overall detection result than each single module could provide. In this case the combination should be made among scratch masks B^j produced by the detection modules, $j = 1, \ldots, d$ (in our case $d = 3$). Here, two different combining methods or aggregation operators are adopted; supposing, for simplicity, that damaged images are affected by just one scratch, their result B^C is given by:

- *Union aggregation operator*: $B^C = \cup\{B^j : j = 1, \ldots, d\}$ such that $B^C(x, y) = \max_j\{B^j(x, y)\}$, $\forall(x, y)$;
- *Maximum Covering (MC) aggregation operator)*: $B^C = MC\{B^j : j = 1, \ldots, d\}$ such that, for all $y = 0, \ldots, M - 1$ (M = number of image rows),

$$B^C(x, y) = \begin{cases} 1 \text{ if} x \in [x^{mean} - W, x^{mean} + W] \\ 0 \text{ otherwise} \end{cases},$$

where $W = \max\{|x^{mean} - x^{min}|, |x^{mean} - x^{max}|\}$, $x^{mean} = \text{mean}(X)$, $x^{min} = \min(X)$, $x^{max} = \max(X)$, $X = \{x : \cap_j B^j(x, y) = \min_j\{B^j(x, y)\} = 1, \forall y\}$.

2.2 Removal Fusion Strategies

The problem here can be stated as follows: given r images representing heterogeneous data on the observed phenomenon, take a decision D_i on an element (x, y) where D_i belongs to a decision space D. In image fusion the information relating (x, y) to each possible decision D_i is represented as a number M_i^j, where j indexes the decision making module having different properties and different meanings depending on the mathematical framework. Given images R^j obtained by removal module j, $j = 1, \ldots, r$ (in our case $r = 2$), if we assume

that $M_i^j(x,y) = R^j(x,y)$, with (x,y) in the scratch domain, represents the probability degree to which the pixel (x,y) could be seen as "restored" (i indexes the values of this appearance), we can claim all the advantages of the Bayesian framework relying in the variety of combination operators. Here we adopt the averaging aggregation operator, known in the Bayesian framework as the Basic Ensemble Method [17] (see also [18,19]) for combining different classification modules, which has been demonstrated to significantly improve the classification performance of each single module: $R^C = BEM\left\{R^j : j = 1,\ldots,r\right\}$, such that $R^C(x,y) = \frac{1}{r}\sum_{j=1}^{r} R^j(x,y)\ \forall(x,y)$.

3 Qualitative Results

3.1 Test Data

Detection and removal algorithms here presented have been tested on several real images. Moreover, the considered algorithms have been tested also on artificially corrupted images. Specifically we considered $K = 20$ uncorrupted original B/W images $I^k, k = 1,\ldots,K$, each of size $N \times M = 256 \times 256$, and the corresponding images with an artificial scratch of odd width w, denoted as $I^{k,w}, k = 1,\ldots,K; w = 3,5,\ldots,19$, obtained as:

$$I^{k,w}(x,y) = \begin{cases} 255 & \text{if}(x,y) \in \Omega_w - \partial\Omega_w \\ 200 & \text{if}(x,y) \in \partial\Omega_w \\ I^k(x,y) & \text{otherwise} \end{cases}, \tag{2}$$

where Ω_w denotes the scratch domain, that is the rectangular subset of the image domain of size $w \times M$ having as first column the center column $N/2$ of the image: $\Omega_w = \left\{(x,y) : x = \frac{N}{2},\ldots,\frac{N}{2}+w-1; y = 0,\ldots,M-1\right\}$, and $\partial\Omega_w$ denotes its border.

3.2 Detection Results

The accuracy of the result of the detection algorithms taken into account (not reported here for space constraints) is quite high. Nonetheless, the aggregated masks seem more appropriate for the successive removal phase. Comparing the two aggregated masks, the union aggregation operator allows to consider all details captured by the different detection algorithms, while retaining the minimum support of the mask; the MC aggregation operator, instead, leads to a mask that appears unnatural for real images (being perfectly rectangular).

In order to obtain an objective estimate of detection algorithms, for each mask B^k computed with anyone of the described detection algorithms for the artificially scratched image $I^{k,w}$ described in eqn. (2) we count:

$C^{k,w} = card\left\{(x,y) : (x,y) \in \Omega_w, B^k(x,y) = 1\right\}$, number of *correct detections* (pixels of the scratch that are included in the computed scratch mask);

Table 1. *Correct detection rates* r_C and *false alarm rates* r_F for the detection algorithms applied to images of eqn. (2) varying the scratch width w.

	$w=3$		$w=5$		$w=7$		$w=9$	
	r_C	r_F	r_C	r_F	r_C	r_F	r_C	r_F
Detection alg. 1	0.9984	0.0000	0.9991	0.0000	0.9991	0.0000	0.9991	0.0000
Detection alg. 2	0.9684	0.0006	0.9565	0.0015	0.9697	0.0017	0.9765	0.0017
Detection alg. 3	0.9984	0.0000	0.9991	0.0000	0.9996	0.0000	0.9995	0.0000
Union fusion aggr. op.	0.9984	0.0006	0.9991	0.0015	0.9996	0.0017	0.9995	0.0017
MC fusion aggr. op.	1.0000	0.0012	1.0000	0.0024	1.0000	0.0024	1.0000	0.0028

$F^{k,w} = card\{(x,y) : (x,y) \notin \Omega_w, B^k(x,y) = 1\}$, number of *false alarms* (pixels not belonging to the scratch that are included in the computed scratch mask),

and their mean values C^w and F^w over the K images of eqn. (2): $C^w = \frac{1}{K}\sum_{k=1}^{K} C^{k,w}$; $F^w = \frac{1}{K}\sum_{k=1}^{K} F^{k,w}$. Given the scratch width w, the measures adopted for the objective estimation of the detection algorithms are:

- *correct detection rate* $r_C = \frac{C^w}{w \times M}$, $w \times M$ being the number of corrupted pixels (i.e. the dimension of the set Ω_w). Such measure gives values in [0,1]; the higher the value of r_C, the better the detection result;
- *false alarm rate* $r_F = \frac{F^w}{N \times M - w \times M}$. Such measure gives values in [0,1]; the lower the value of r_F, the better the detection result.

Values for r_C and r_F obtained with all the described detection algorithms are reported in Table 1, varying the scratch width w =3, 5, 7, and 9. Here we can observe that r_C values are generally very close to 1 for all detection algorithms and that only few false alarms are generated. Specifically, we observe that the union fusion strategy reaches the best r_C and the worst r_F values achieved by the underlying detection algorithms; the MC fusion strategy reaches the best r_C values attainable ($r_C = 1$), but r_F values worse than any other algorithm.

3.3 Removal Results

The results of scratch removal (not reported here for space constraints) show that, even though the removal algorithms taken into account perform quite well, their reconstruction accuracy can be enhanced; the aggregated results, instead, tend to smooth the inaccuracies, still retaining the good performance of the considered algorithms.

In order to obtain objective measures of the removal algorithms accuracy, we tested them on the artificially scratched images $I^{k,w}$ described in eqn. (2). Given the scratch width w, let be, for $k = 1, \ldots, K$, $\mathbf{o}_k$ the vector of dimension $card(\Omega_w) \times 1$ obtained scanning row by row the subimage of I^k for pixels in Ω_w, and $\mathbf{r}_k$ the vector of dimension $card(\Omega_w) \times 1$ obtained scanning row by row the subimage of R^k for pixels in Ω_w. We consider the following objective measures:

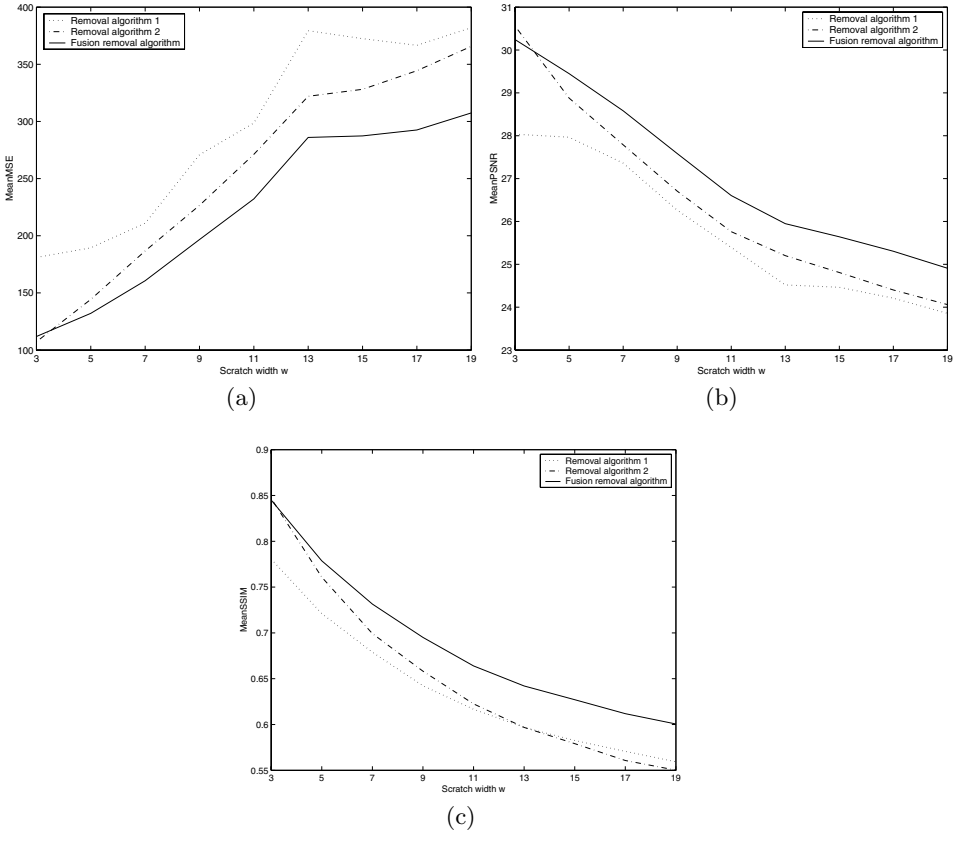


Fig. 1. Accuracy measures of the removal algorithms applied to images described in eqn. (2): (a) *MeanMSE*, (b) *MeanPSNR*, (c) *MeanSSIM*

- $MeanMSE = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{w*M}\left\|\mathbf{o}_k - \mathbf{r}_k\right\|^2$, mean, over the K restored images, of the Mean Square Error (MSE) between the original and the restored images. Such measure gives a nonnegative value; the smaller the value of $MeanMSE$, the better the restoration result;
- $MeanPSNR = \frac{1}{K}\sum_{k=1}^{K}\left(10 * log_{10}\left(\frac{255^2}{\frac{1}{w*M}\left\|\mathbf{o}_k - \mathbf{r}_k\right\|^2}\right)\right)$, mean, over the K restored images, of the Peak-to-Noise-Ratio between the original and the restored images obtained considering the MSE. Such measure gives a nonnegative value; the higher the value of $MeanPSNR$, the better the restoration result;
- $MeanSSIM = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\left(2 * \mu_{\mathbf{o}_k} * \mu_{\mathbf{r}_k} + C_1\right)\left(2 * \sigma_{\mathbf{o}_k\mathbf{r}_k} + C_2\right)}{\left(\mu_{\mathbf{o}_k}^2 + \mu_{\mathbf{r}_k}^2 + C_1\right)\left(\sigma_{\mathbf{o}_k}^2 + \sigma_{\mathbf{r}_k}^2 + C_2\right)}\right)$, mean, over the K restored images, of the Structural Similarity Index [20] applied to the

original and the restored images, where $C_1 = (K_1 * L)^2, C_2 = (K_2 * L)^2, K_1 = 0.01, K_2 = 0.03$, and $L = 255$. Such measure gives values in [0,1]; the higher the value of $MeanSSIM$, the better the restoration result.

Results obtained with the described measures varying the scratch width w are reported in Fig. 1. It can be observed that $MeanMSE$ values obtained with the fusion removal algorithm are always lower than those obtained with the two removal algorithms and that $MeanPSNR$ and $MeanSSIM$ values obtained with the fusion removal algorithm are always higher than those obtained with the two removal algorithms. Moreover, for each removal method, results obtained with all the considered measures show lower accuracy increasing the scratch width, in accordance with the increasing reconstruction difficulty as the reconstruction area widens.

In summary, we can state that the considered measures indicate the fusion method as the most accurate among the considered removal methods.

4 Conclusions and Ongoing Work

This paper described an innovative algorithm for line scratch restoration based on data fusion techniques to detect and restore scratches in digital corrupted images. The described compound algorithm has been tested on several corrupted and artificially corrupted real images in order to analyze the results accuracy in terms of objective measures, showing that the compound algorithm outperforms the underlying restoration methods.

Ongoing researches deal with the analysis and adoption of alternative aggregation operators, like the Ordered Weighted Aggregation operators due to Yager [21] or the Non-liner Generalized Ensemble Method as introduced in [22].

References

1. Acton, S.T., Mukherjee, D.P., Havlicek, J.P., Bovik, A.C.: Oriented Texture Completion by AM-FM Reaction-Diffusion. IEEE Transactions on Image Processing **10** (2001) 885-896
2. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image Inpainting. Computer Graphics (2000) 417-424
3. Bornard, R., Lecan, E., Laborelli, L., Chenot, J.-H.: Missing Data Correction in Still Images and Image Sequences. In Proc. ACM Multimedia 2002, Juan-les-Pins, France (2002) 355-361
4. Chan, T.F., Shen, J.: Mathematical Models for Local Non-Texture Inpaintings. UCLA CAM Report n. 00-11 (2000)
5. Decenciere Ferrandiere, E.: Restauration Automatique de Films Anciens. PhD Thesis, Ecole Nationale Superieure des Mines de Paris (1997)
6. Isgró, F., Tegolo, D.: A distributed genetic algorithm for restoration of vertical line scratches. Accepted for publication in Parallel Computing
7. Joyeux, L., Boukir, S., Besserer, B.: Tracking and MAP Reconstruction of Line Scratches in Degraded Motion Pictures. Machine Vision and Applications **13** (2002) 119-128

8. Kao, O., Engehausen, J.: Scratch Removal in Digitised Film Sequences. In Proc. International Conference on Imaging Science, Systems, and Technology (CISST) (2000) 171-179
9. Kokaram, A.C.: Motion Picture Restoration: Digital Algorithms for Artefacts Suppression in Archived Film and Video. Springer-Verlag (1998)
10. Machì A., Collura, F., Nicotra, F.: Detection of Irregular Linear Scratches in Aged Motion Picture Frames and Restoration using Adaptive Masks. In Proc. IASTED Int. Conf. SIP02, Kawai, Usa (2002) 254-259
11. Maddalena, L.: Efficient Methods for Scratch Removal in Image Sequences. In Proc. 11th International Conference on Image Analysis and Processing (ICIAP2001), IEEE Computer Society (2001) 547-552
12. Morris, R.D.: Image Sequence Restoration Using Gibbs Distributions. PhD Thesis, University of Cambridge (1995)
13. Saito, T., Komatsu, T., Ohuchi, T., Seto, T.: Image Processing for Restoration of Heavily-Corrupted Old Film Sequences. In Proc. ICPR'00, IEEE, Barcellona (2000) 3017-3020
14. Bloch, I.: Information Combination Operators for Data Fusion: A Comparative Review with Classification. IEEE Transactions on Systems, Man, Cybernetics **26** (1996) 52-67
15. Ho, T.K., Hull, J.J., Srihari, S.N.: Decision Combination in Multiple Classifier Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence **18** (1994) 66-75.
16. Laccetti, G., Maddalena, L., Petrosino, A.: Parallel/Distributed Film Line Scratch Restoration by Fusion Techniques. In: Lagan, A. et al. (eds.): Computational Science and Its Applications - ICCSA 2004, Lecture Notes in Computer Science, Vol. 3044. Springer (2004) 524-534
17. Perrone, M.P., Cooper, L.N.: When networks disagree: Ensemble method for neural networks. In Mammone, R.J. (ed.): Artificial Neural Networks for Speech and Vision. Chapman & Hall, New York (1993) 126-142
18. Roli, F.: Linear Combiners for Fusion of Pattern Classifiers. Int. School on Neural Nets, E.R. Caianiello, 7th Course on Ensemble Methods for Learning Machines, Vietri sul Mare, Italy (2002)
19. Fumera, G., Roli, F.: A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence. (in press)
20. Wang, Z., Lu, L., Bovik, A.C.: Video Quality Assessment Based on Structural Distortion Measurement. Signal Processing: Image Communication **19** (2004) 121-132
21. Yager, R.R., Kacprzyk, J.: The Ordered Weighted Averaging Operation: Theory, Methodology and Applications. Kluwer: Norwell, MA (1997)
22. Ceccarelli, M., Petrosino, A.: Multifeature adaptive classifiers for SAR image segmentation. Neurocomputing **14** (1997) 345-363

Time and Date OCR in CCTV Video

Ginés García-Mateos[1], Andrés García-Meroño[1], Cristina Vicente-Chicote[3], Alberto Ruiz[1], and Pedro E. López-de-Teruel[2]

[1] Dept. de Informática y Sistemas
[2] Dept. de Ingeniería y Tecnología de Computadores,
Universidad de Murcia, 30.170 Espinardo, Murcia (Spain)
[3] Dept. Tecnologías de la Información y Comunicaciones,
Universidad Politécnica de Cartagena, 30.202 Cartagena, Murcia (Spain)
{ginesgm, aruiz, pedroe}@um.es, andres.garcia2@carm.es,
cristina.vicente@upct.es

Abstract. Automatic recognition of time and date stamps in CCTV video enables the inclusion of time-based queries in video indexing applications. Such ability needs to deal with problems of low character resolution, non-uniform background, multiplexed video format, and random access to the video file. In this paper, we address these problems and propose a technique that solves the difficult task of character segmentation, by means of a recognition-based process. Our method consists of three main steps: pattern matching, character location and syntactic analysis. The experiments prove the reliability and efficiency of the proposed method, obtaining an overall recognition rate over 80%.

1 Introduction and Related Research

Optical character recognition in digital video (in short, video OCR) has received an increasing interest in the last few years [1]. Some of its applications include video indexing and digital libraries [2,3,4], commercial segments detection [2], sign detection and recognition [6], MPEG-7 text descriptions and video summarization [5]. In this paper, we address the problem of time and date stamp recognition in CCTV (Closed Circuit TV) surveillance video. While the problem is still in the scope of video OCR, some peculiarities suggest the development of specific methods, as we will discuss.

Although digital CCTV systems will eventually supersede analogic CCTV, at present, analogic systems are far more commonly used in supermarkets, banks, traffic control, and similar applications. A typical CCTV system consists of several cameras, placed either indoors or outdoors. Their outputs are multiplexed and recorded in a single analogic videotape, i.e. successive frames correspond to different cameras, as shown in Fig. 1. Additionally, the multiplexer adds to each frame a time and date stamp, and optionally the camera number.

The video OCR presented here was designed to be integrated in a bigger system[1], which is able to digitize, demultiplex and process analogic tapes of this

[1] Owned by Vision Base Int. Ltd. We want to thank their support in this research.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 703–710, 2005.

Fig. 1. Sample time and date stamps in consecutive frames. Image resolution is 704x286 pixels (1 image = 1 field). Input video frequency is 4 fps.

kind of systems. The aim of our work is to add time and date stamps recognition functionality, allowing time-based queries. These queries could be of the form: "select the frames between time t_1 and t_2", "move to instant t", or "show only the frames from camera number n".

It is widely accepted that the two main problems in video OCR are complex background and low resolution [1,2]. The former complicates segmentation, since character and background pixels have similar values, while the later means that recognition is very sensitive to noise. Time and date OCR suffers from both problems, and has to deal with two added difficulties. First, the video is multiplexed, which involves the lack of continuity between consecutive frames. Second, queries require a random access to any part of the video. Considering also the low input frequency –which in CCTV systems is usually from 1 to 5 fps–, we adopt the premise that each frame has to be recognized individually.

On the other hand, the text to recognize is not a mere sequence of characters, but a valid date and time. Different syntaxes of time and date have to be considered, which include: time and date in one or two lines; date above or below; numeric or alphabetic month representation; seconds with one decimal digit; order of day and month in the date; different separators, etc.

The rest of the paper is organized as follows. Section 2 gives a general overview of the method. Next, we describe our solution in sections 3, 4 and 5, detailing the three main steps: pattern matching, character location and syntactic analysis. Finally, we present some experimental results and conclusions.

2 System Overview

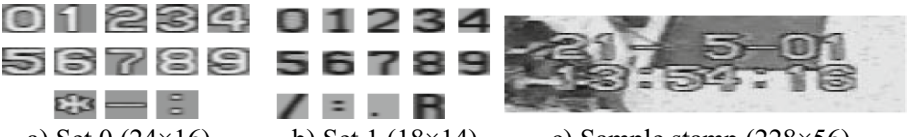
Time and date OCR in CCTV video has to cope with high variability in background, location of the stamp in the image, font type, and syntactic format. However, in a single video sequence all of them, except background, suffer no change. Thus, we decompose the problem into detection and updating. In the detection phase, a costly search through all possible locations, fonts and formats is applied to select the most likely combination[2]. After that, the updating phase performs an easy and efficient computation of the current time and date.

[2] By application requirement, a region of interest (ROI) is supposed to be manually selected by the user in the images.

Since segmentation is not feasible under the existing background conditions, we propose a method which does not require prior segmentation; it can be considered a case of recognition-based segmentation [2], where segmentation takes place only after a pattern matching process. The detection phase consists of three main steps. First, pattern matching is applied in order to detect the characters of the stamp. Second, characters are located and arranged in a string, according to the maximum values of matching. Third, syntactic analysis is performed, selecting the most likely representation of time and date among a predefined set of valid formats.

3 Pattern Matching

Time and date stamps are superimposed to the video signals using a reduced number of standard font types, specific of each manufacturer. Fig. 2 shows two of these types. Contrary to printed text OCR, neither rectification nor scale are needed, as font size does not change and the baseline is always horizontal. Under both conditions, we can apply a simple pattern matching on the ROI, to detect characters of the predefined font types.



a) Set 0 (24×16) b) Set 1 (18×14) c) Sample stamp (228×56)

Fig. 2. Two font sets and a sample time and date stamp. Fonts and stamp sizes (width×height) are indicated below.

Patterns of the characters are compared through all the ROI using a normalized coefficient matching, defined as follows. Let i and t be the image and the pattern, respectively, and let t be of size $w \times h$. The normalized patches of i and t are given by:

$$i'(x,y) = i(x,y) - \frac{1}{wh} \sum_{a=0}^{w-1} \sum_{b=0}^{h-1} i(x+a, y+b) \tag{1}$$

$$t'(x,y) = t(x,y) - \frac{1}{wh} \sum_{a=0}^{w-1} \sum_{b=0}^{h-1} t(a,b) \tag{2}$$

That is, both i' and t' have zero mean brightness. Matching value is a cross correlation of these normalized patches:

$$m_{i,t}(x,y) = \frac{\sum_{a=0}^{w-1} \sum_{b=0}^{h-1} t'(a,b) i'(x+a, y+b)}{\sqrt{\sum_{a=0}^{w-1} \sum_{b=0}^{h-1} t'(a,b)^2 \sum_{a=0}^{w-1} \sum_{b=0}^{h-1} i'(x+a, y+b)^2}} \tag{3}$$

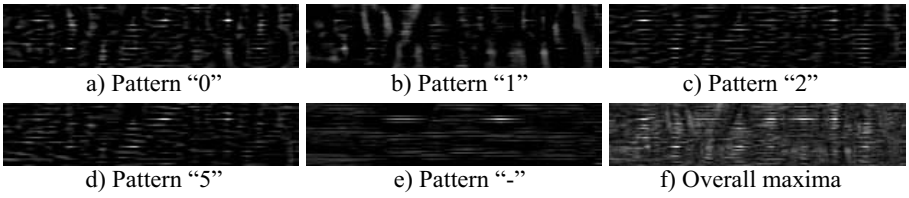


Fig. 3. Matching results of stamp in Fig. 2c) with some patterns of the set shown in Fig. 2a). Time and date digits are superimposed on their original locations, for visualization purposes. White means a better matching, and it should be located on the upper-left corner of the characters.

Fig. 3 presents some sample results of this step. In general, local maxima appear at the expected positions. However, we can observe that many spurious local maxima exists, thus complicating the task of character location.

Obviously, pattern matching is the most time-consuming step of the process. In the detection phase –when no information concerning time and date is assumed–, all patterns from all font types are matched with the image. In the following frames, only a few patterns are matched in small parts of the ROI, as far as font type and locations of characters are known.

4 Character Location and Segmentation

After the pattern matching step, we obtain a set of *matching maps*, $m_{i,t}$, indicating the likelihood that each character t is present at every possible location of the ROI. However, extracting a coherent string of characters is not a trivial task. As we have seen in Fig. 3, most of the existing local maxima do not correspond to real character locations. Moreover, some character patterns produce very low maxima even in the correct locations; see, for example, number "1" in Fig. 3b).

We propose a method to cope with both problems, producing a string of locations for a given font set. The resulting string is stored in a matrix[3], p, of size $\lfloor W/w \rfloor \times \lfloor H/h \rfloor$, where $W \times H$ is the ROI size, and $w \times h$ is the font size. Our method is based on the following greedy algorithm.

Let us consider a font type s consists of a set of character patterns, $s = \{s_0, s_1, ..., s_9, s_{10}, ..., s_k\}$, where s_0 to s_9 are the corresponding decimal digits, and s_{10} to s_k are separators. All these patterns have the same size, $w \times h$. We postulate that the maximum matching through all possible locations and patterns in s is a correct recognition. That is, the upper-left corner of the first segment is:

$$(x_0, y_0) = \arg \max_{\forall x,y} \{ \max_{\forall c} (m_{i,s_c}(x, y)) \} \tag{4}$$

The assumption that (x_0, y_0) is a correct segment, is the starting point of the algorithm. This way, cell $p(a, b)$ is set to (x_0, y_0), with $a = \lfloor x_0/w \rfloor$ and $b =$

[3] Recall that time and date can appear in one or two rows.

$\lfloor y_0/h \rfloor$. Since all characters are equally spaced, the left adjacent segment, $p(a-1,b)$, should be located in (x_0-w, y_0), and the right one in (x_0+w, y_0). However, this fixed jump of w pixels could accumulate small errors, producing a bad segmentation of distant characters. Thus, we compute the maximum matching in a certain tolerance region, of size $r_x \times r_y$:

$$(x_1, y_1) = \arg \max_{\forall x \in R_x, \forall y \in R_y} \{\max_{\forall c}(m_{i,s_c}(x,y))\} \tag{5}$$

with:

$$R_x = \{x_0 + w - r_x, ..., x_0 + w + r_x\}; R_y = \{y_0 - r_y, ..., y_0 + r_y\} \tag{6}$$

for the right adjacent character. Then, segment $p(a+1,b)$ is given by:

$$p(a+1,b) = \left\{ \begin{array}{cc} (x_1, y_1) & if \max_{\forall c}(m_{i,s_c}(x_1,y_1)) > threshold \\ (x_0+w, y_0) & otherwhise \end{array} \right\} \tag{7}$$

That is, the maximum is not considered if it is below a given threshold; instead, the predefined width w is used. This condition is necessary to avoid low spurious maximums that appear in the place of blank spaces (see an example in the month field in Fig. 2c). The process is repeated to the left, to the right, and then above and below, until the matrix of locations p is completed. Fig. 4 shows the results of this algorithm when applied on the stamp in Fig. 2c).

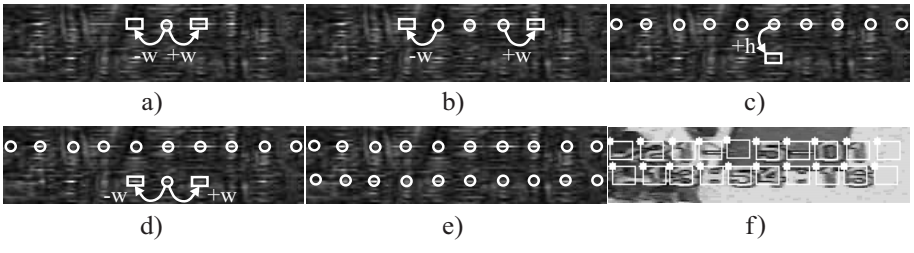


Fig. 4. Character location and segmentation algorithm. Starting from the overall maximum matching, a), the process moves left and right searching for local maxima, b) and c), until it reaches the extremes. Then, we move below and repeat the process, d) and e). The resulting segmentation is shown in f).

5 Syntactic Analysis

The output of a generic OCR would basically consist of taking the characters that produce the highest matching for every segment in string p. But, in our case, syntactic analysis is required to enforce an interpretation of the result as a valid time and date. Two problems are involved: *where* are located time and date in p, and *what* time and date format is used. We consider a predefined set of

valid time and date formats. Syntactic analysis consists of searching the format that better fits in the computed string, according to the matching values.

Each time and date format is described as a string of format elements, where a format element represents a set of valid characters. We have also defined *double* format elements, that stand for sets of valid pairs of characters –e.g., any pair from "00" to "23"–. We distinguish 6 single and 6 double format elements, as shown in Table 1. Using these format elements, three possible time formats could be "[00-23]:59", "[00-23]:59:59" and "[00-23]:59:59.9". In the first case, only hour and minutes appear, while the third represents seconds with one decimal digit.

Table 1. Format elements used to represent dates and times. Left: a single element stands for a set of valid characters. Right: a double element stands for a valid pair of characters.

Single element	Set of characters, $set(.)$										Double element	Corresponding pairs of characters
	blank	0	1	2	3	4	5	6	7	8	9	
"*"	x											"[0-23]"	*0, *1, *2, ..., 23
"1"		x	x									"[00-23]"	00, 01, 02, ..., 23
"2"		x	x	x								"[1-12]"	*1, *2, *3, ..., 12
"3"		x	x	x	x							"[01-12]"	01, 02, 03, ..., 12
"5"		x	x	x	x	x	x					"[1-31]"	*1, *2, *3, ..., 31
"9"		x	x	x	x	x	x	x	x	x	x	"[01-31]"	01, 02, 03, ..., 31

We define the likelihood that each character, single or double element is present at every segment –or pair of segments, in the last case– of a string. The likelihood of a particular character t in position a of the string is given by $l(t,a) = m_{i,t}(p(a,0))$, assuming the stamp occupies one single row. The likelihood of a format element e is given by $l(e,a) = max_{t \in set(e)} l(t,a)$, where $set(e)$ is the set of characters associated to e, as shown in Table 1. We pinpoint the special case of the blank space, whose likelihood is $l(\text{“}*\text{”}, a) = 1 - max_{\forall t} l(t,a)$. Finally, the likelihood of a double format element accounts for all possible combinations of pairs; for example, the likelihood of "[0-23]" is given by $l(\text{“}[0-23]\text{”}, a) = max\{l(\text{“}*\text{”}, a) + l(\text{“9”}, a+1), l(s_1, a) + l(\text{“9”}, a+1), l(s_2, a) + l(\text{“3”}, a+1)\}$.

Moveover, the likelihood that a time or date format, f, is present in p starting from position a can be easily defined. If f is composed of elements $f(1)$, $f(2)$, ..., $f(n)$, the likelihood of f is:

$$l(f,a) = \sum_{i=1}^{n} l(f(i), a+i-1) \tag{8}$$

In this way, syntactic analysis consists of finding the format f and the location a that maximize equation 8. Once a format is selected, we have semantic information of every segment in p, i.e., where the hour, minutes, seconds, etc., are. Finally, we simply take the characters with the highest likelihoods, and coherent with the format element in each position.

6 Experimental Results

We have evaluated our method using a set of 17 CCTV videos at a 704×286 resolution. The total number of images in these videos is 5337. Although the video frequency is 25 fps, the typical input frequency is about 5 fps, so more than 1000 different times and dates are available for testing. We have trained and used 9 font sets, applying a semi-automatic training process (using a temporal average of the characters and manual segmentation). On the other hand, 9 different time and date formats were defined and used in the experiments.

Table 2 summarizes the results of our method. For each video sequence, the three steps of the process –pattern matching, character location and syntactic analysis– are applied to the first frame in the selected ROIs, using all the font sets and formats. This is the detection phase, as described in Section 2. In the rest of frames, pattern matching is restricted to the found font type, in the previously selected segments, and with patterns that are admissible in each segment –according to the selected format–. This is what we call the updating phase. For clarity reasons, the results of the 17 videos are joined into 4 groups, from the most favorable (group 1) to the most difficult cases (group 4).

Table 2. Recognition results of the proposed method. The experiments have been done on an AMD Athlon XP 2000+ with 256 Mbytes of RAM.

	Number of frames	% Correct characters	Correct stamps	% Correct stamps	Detection time (ms)	Updating time (ms)
Group 1	1522	99.9%	1512	99.3%	814.1	8.7
Group 2	1318	99.3%	1220	92.6%	1031.1	30.2
Group 3	1594	95.9%	1186	74.4%	794.6	17.8
Group 4	903	90.8%	380	42.1%	912.2	8.5
Total	5337	97.0%	4298	80.5%	889.5	16.6

As shown in Table 2, we achieve a very high character recognition rate of 97%, despite the problems described in Section 4. Most of the errors are due to two reasons: confusion between similar patterns, like "5" and "6", and low matching values for patterns with many background pixels, like "1" and "-".

Stamp recognition rates show a higher variability from one group to another, with an overall rate of 80.5%. A successful result does not necessarily involve a correct recognition of all characters –that could be corrected in the syntactic analysis–. However, a single error in a decimal digit is prone to cause an incorrect output. This explains the low ratio of 42.1% in group 4, which presents severe noise conditions. Fig. 5 shows some correct and wrong results.

7 Conclusions

Despite the increasing need for general text recognition in video [2], OCR in video is still a challenging problem. In this paper we have addressed the case

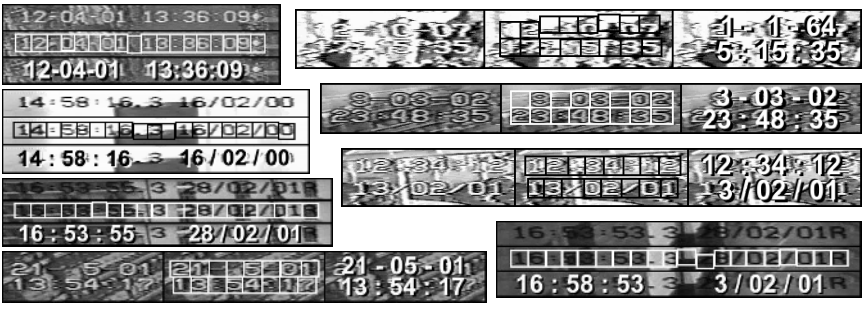


Fig. 5. Some sample results. Left: correct recognition of time and date. Right: incorrect results. For each stamp, character segmentation and system outputs are shown.

of time and date stamp recognition in CCTV video. While, in our case, some specific restrictions simplify the complexity of the problem, we have to deal with two additional difficulties: multiplexed video format and random file access.

We have developed a method which does not require character segmentation prior to recognition. First, pattern matching is applied in the ROI of the images, and then a greedy algorithm locates the characters using matching values. Finally, we perform a syntactic analysis step that selects the most likely presence of a valid time and date in the located segments.

Time and date OCR in CCTV video provides useful information that cannot be obtained by other means. The integration of our technique in a bigger application will offer the ability of performing time-based queries. The experimental results exhibit high recognition rates, showing the feasibility and computational efficiency of our approach.

References

1. Lienhart, R.: Video OCR: A Survey and Practitioner's Guide. Video Mining. Kluwer Academic Publisher (2003) 155–184
2. Sato, T., Kanade, T., Hughes, E.K., Smith, M.A., Satoh, S.: Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions. ACM Multimedia Systems, Vol. 7, No. 5 (1999) 385–395
3. Jain, A.K., Yu, B.: Automatic Text Localication in Images and Video Frames. Pattern Recognition, 31(12) (1998) 2055–2076
4. Li, H., Doermann, D., Kia, O.: Automatic Text Detection and Tracking in Digital Video. IEEE Trans. on Image Processing, 9(1) (2000) 147–156
5. Lienhart, R., Pfeiffer, S., Effelsberg, W.: Video Abstracting. Communications of the ACM, Vol. 40, No. 12 (1997) 55–62
6. Yang, J., Chen, X., Zhang, J., Zhang, Y., Waibel, A.: Automatic Detection and Translation of Text from Natural Escenes. Proc. of ICASSP'02, vol. 2 (2002)

Statistical Modeling of Huffman Tables Coding

S. Battiato[1], C. Bosco[1], A. Bruna[2], G. Di Blasi[1], and G. Gallo[1]

[1] D.M.I. – University of Catania - Viale A. Doria 6,
95125, Catania, Italy
{battiato, bosco, gdiblasi, gallo}@dmi.unict.it
[2] STMicroelectronics - AST Catania Lab
arcangelo.bruna@st.com

Abstract. An innovative algorithm for automatic generation of Huffman coding tables for semantic classes of digital images is presented. Collecting statistics over a large dataset of corresponding images, we generated Huffman tables for three images classes: landscape, portrait and document. Comparisons between the new tables and the JPEG standard coding tables, using also different quality settings, have shown the effectiveness of the proposed strategy in terms of final bit size (e.g. compression ratio).

1 Introduction

The wide diffusion of imaging consumer devices (Digital Still Cameras, Imaging Phones, etc.) coupled with the increased overall performances capabilities, makes necessary an increasing reduction in the bitstream size of compressed digital images. The final step of the JPEG baseline algorithm ([5]) is a lossless compression obtained by an entropy encoding. The normalized DCT coefficients, properly grouped according to some simple heuristics techniques, are encoded by classical Huffman coding ([3],[11]) making use of a set of coding tables. Usually, the standard coding tables are used, as suggested in ([5], [8]) and included in the final bit-stream for every image to be compressed. This approach presents two main disadvantages:

1. the JPEG Huffman encoder writes all codes of the corresponding tables in the final bitstream even if only some of them were used to encode the associated events of the particular input image. An overhead (mainly for high compression rate) is clearly present because the header of the JPEG file will contain complete tables where unused codes are stored.
2. besides the JPEG standard encoder doesn't make use of any statistic about the distribution of the events of the current image.

To overcome these problems the JPEG encoder can be modified so that it can compute the frequencies of the events in each image to encode as described in [5]. The implementation of these algorithms allows obtaining an Huffman optimizer that, as a black box, takes as input the frequencies collected for a single image and generates optimal coding tables for it.

As shown in Figure 1, the optimizer requires a pre-processing phase in which the statistics of the current image are collected. It's not always possible implementing

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 711 – 718, 2005.

such computation in embedded systems for low-cost imaging devices, where limited resources are available. On the other hand, static Huffman coding can be properly managed collecting statistics for a class of data having relatively stable characteristics. This approach is also used in [7] where experiments with a previous phase of collection of statistics for source programs in four programming languages are successfully applied.

In this work we present an algorithm to generate Huffman coding tables for classes of images conceived using the Huffman optimizer. Applying the new generated tables instead of the standard ones, it's possible to obtain a further reduction in the final size of the bitstream without loss of quality.

New tables for three classes of images (landscape, portrait and document) were generated taking into account statistical similarities, measured in the "event space". For these particular classes psychovisual and statistical optimization of DCT quantization tables was presented in [1]. In [6] a technique to achieve an improved version of the JPEG encoder modifying the Huffman algorithm is presented; nevertheless using multiple tables to have better compression generates a bitstream not fully compliant with the standard version of the JPEG. Similar drawbacks can be found in ([2], [10]). Our proposed approach allows obtaining a standard compliant bitstream.

The paper is structured as follows: the next section describes the new proposed algorithm; section III reports the experimental results while in the next section a few words about statistical modeling of frequency distribution are reported. Finally, a brief conclusive section pointing to future evolutions is also included.

Fig. 1. The Huffman Optimizer

2 Algorithm Description

After the quantization phase, the JPEG encoder manages the 8x8 DCT coefficients in two different ways: the quantized DC coefficient is encoded as the difference from the DC term of the previous block in the encoding order (differential encoding) while the quantized AC coefficients are ordered into a "zig-zag" pattern to be better managed by entropy coding. In the baseline sequential codec the final step is the Huffman coding. The JPEG standard specification ([5]) suggests some Huffman tables providing good results on average, taking into account typical compression performances on different kinds of images. The encoder can use these tables or can optimize them for a given image with an initial statistics-gathering phase ([8], [11]). The Huffman optimizer is an automatic generator of dynamic optimal Huffman tables for each single image to encode by JPEG compression algorithm. If the statistics, taken as input by the optimizer block, refer to a dataset of images rather than to a single image, the optimizer generates new Huffman coding tables for the global input dataset (as shown in Figure 2).

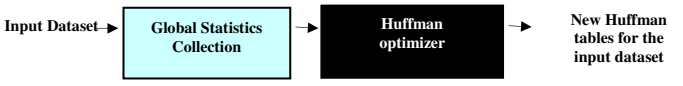


Fig. 2. A block description of the steps performed by the algorithm for automatic generation of Huffman coding tables for classes of images

To collect reliable data, the input dataset should contain a large number of images belonging to the same semantic category: we claim that images of the same class have similar distribution of events. The concept of semantic category has been also successfully used in ([1], [7]).

We considered three classes of images as input for our algorithm: landscape, portrait and document. The main steps of the proposed technique can be summarized as follows:

1. The algorithm takes as input a dataset of images belonging to the same semantic category;
2. Each single image of the considered dataset is processed collecting its statistics;
3. The statistics of the currently processed image are added to the global frequencies (i.e. the statistics corresponding to already processed images);
4. After the computation of statistics for the whole dataset, the values of global frequencies that are equal to zero are set to one;
5. In the last step the Huffman optimizer takes as input the global statistics and returns the new Huffman coding tables corresponding to the input dataset.

Step IV guarantees the generation of complete coding tables as well as the standard tables are complete. In the worst case it must be possible to associate a code to each generated event. The Huffman optimizer for single image, showed in Figure 1, doesn't create complete tables because all the events in the image are known, hence just such events must have an Huffman code. The described algorithm generates new Huffman tables that can be used to encode images belonging to each considered class (landscape, portrait and document). It is possible to improve compression performances of JPEG encoder by replacing the standard tables with the new generated tables for each specific class. Through this substitution, the previous collection of statistics is not needed because fixed Huffman tables are used to encode every image belonging to the three considered classes.

3 Experimental Results

The proposed method has been validated through an exhaustive experimental phase devoted to prove the effectiveness of the generated Huffman tables for each considered semantic class. Initially, in the retrieving phase, three default datasets were used with 60 images for each class chosen at different resolution and acquired by different consumer devices.

Our algorithm, starting from the corresponding class, provides as output the new coding tables for the landscape, portrait and document classes. The standard tables were simply replaced by the new generated Huffman tables.

The default datasets were considered as our training set (TS). So, first of all, we used the TS to make a comparison between the standard coding tables and the new optimized coding tables. Three different types of coding were made on our TS:

1. optimized coding: the images were coded using the optimal tables generated by the Huffman optimizer just for single image;
2. standard coding: the images were coded using the standard tables;
3. optimized coding for class: the images were coded using the new Huffman tables generated for each class.

As expected, the best coding is the optimized coding: it represents a lower bound in terms of dimension of the output bitstream because this coding uses optimal tables for each image to encode.

More interesting results concern the comparison between the standard coding and new tables for each class: the results obtained by optimized coding for class are slightly better than results obtained with default tables.

Table 1. Comparison between the file-size obtained with the standard tables and the derived tables for specific class

Training	LD	MD	Gain	σ
Landscape	-1.6%	19.3%	**5.3%**	4.5%
Portrait	-3.1%	10.6%	**5.9%**	3.2%
Document	-1.3%	11.9%	**1.8%**	2.1%

3.1 Experiments on New Dataset

In order to validate the preliminary results just described above, we performed a further test on three new datasets. The datasets for each class are composed by: 97 landscape images, 96 portrait images and 89 document images. These images were also chosen randomly and they were different from the previous collected ones. Using the corresponding tables properly learned on the training set, all images have been coded. In the table II are reported the relative results that confirm the previous analysis. Just for completeness, in Figure 3 are reported three different plots, corresponding to the measured bpp (bit-per-pixel), obtained coding each image of the new dataset, with the corresponding optimized table for portrait class. Similar results have been measured for the other two classes under investigation.

Table 2. Comparison between standard coding and coding with tables for each class over a large dataset of images not belonging to the training set

Dataset	LD	MD	Gain	σ
Landscape	-0.2%	14.2%	**5.8%**	3.8%
Portrait	-4.3%	13.8%	**5.1%**	4.5%
Document	0.02%	7.2%	**4.1%**	1.7%

Table 3. Percentages of improvement for the landscape class with variable QF

QF	Gain	QF	Gain
10	**18.7%**	40	**7.5%**
20	**13%**	60	**4.3%**
30	**9.7%**	70	**2.2%**

3.2 Experiments Using Different Quality Factory

Experiments devoted to understand the behavior of the new Huffman tables for each class when the Quality Factor (QF) assumes different values are here described. The QF is a JPEG parameter which allows choosing the preferred ratio quality-compression, through a direct modification (usually by a multiplicative factor) of the quantization tables. First, a small set of images of the new dataset belonging to the each class, was randomly chosen, just to evaluate in the range [10, 80] the bpp values obtained by coding with optimized tables for the class. As shown in Figure 4, by using specific tables for landscape class is better than using standard tables. Similar results have been measured for the other two classes under investigation. To further validate these results, the

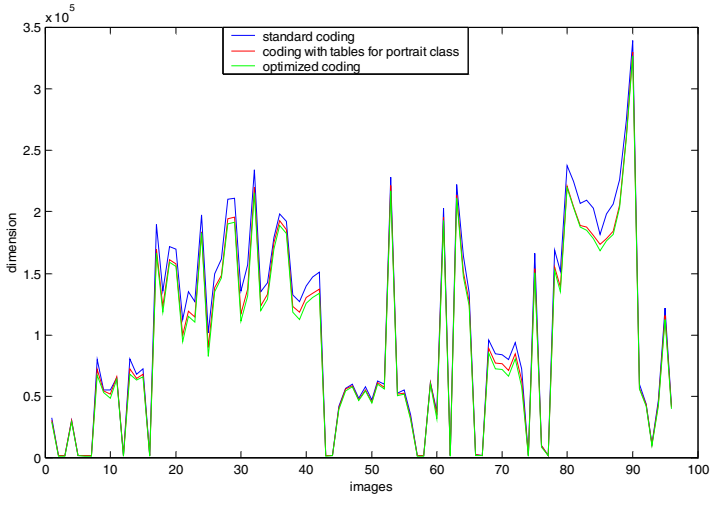


Fig. 3. Three types of coding applied to the new datasets.for portrait images. In all graphics the standard coding is represented by a blue line, the optimized coding for class is represented by a red line while the optimized coding is represented by a green line. In the Y axis of each graphic the dimension is reported in bytes.

three types of coding tables (standard, optimized and optimized for class) were repeated on the overall new dataset (97 landscape images) considering increasing values of the QF. Table III reports the percentages of improvement found for some values of the QF referred to the overall landscape class. Such experiments seem to confirm how

the overall statistics of each class, have been really captured and effectively reported on the Huffman tables. Also the best performances have been noted at lower bit-rate suggesting how there is a sort of regularity among the events generated and coded into the tables.

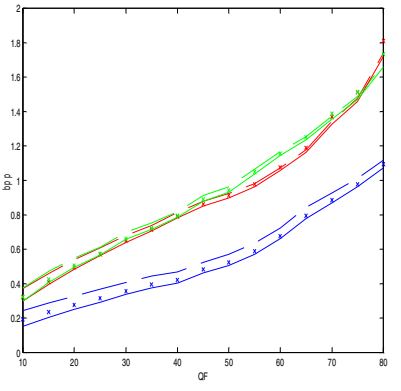


Fig. 4. Three types of coding of three landscape images with variable QF in the range [10, 80]. For each image: the optimized coding is represented by a continuous line, the standard coding is represented by an outlined line and the coding with tables for the landscape class is represented by the symbol X. In the Y axis the bpp (bit per pixel) values are reported while the X axis reports the different values assumed by the QF.

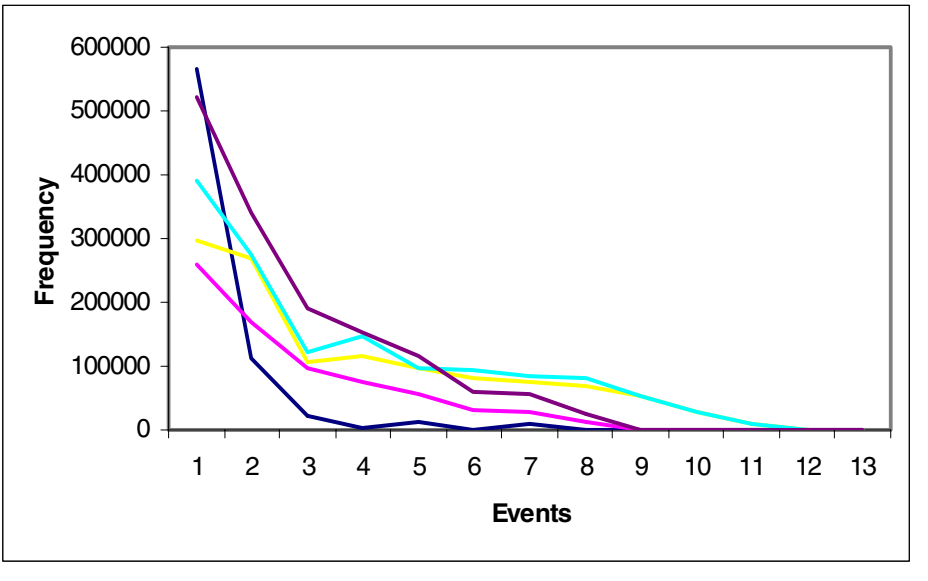


Fig. 5. DC Events frequency distribution for different quality factor values

4 Statistical Modeling

During the experiments performed in the test phase an interesting regularity between quality factor and events frequency has emerged. Such evidence is reported in Figure 5, where the plot corresponding to the DC events frequency distribution for different quality factor values (in the range [0, 100]), just for a single semantic class is showed. We claim that it should be possible fit some mathematical model for these curves capturing the underlying statistical distribution ([4]). In our experiments we have chosen linear spline in order to model the curves with a suitable mathematical function. In this way, just through a homotopyc function between linear spline curves it is possible to adapt the corresponding tables coding for each event frequency distribution for a given QF, without using the statistical classification step described in the previous sections.

Preliminary results seem to confirm our conjecture and suggest us to continue to investigate in this direction; detailed experiments will be reported in the final paper.

5 Conclusions and Future Works

In this paper we presented a new algorithm for automatic generation of Huffman tables based on semantic classes of digital images. By the proposed algorithm we generated new fixed Huffman tables for the following classes: landscape, portrait and document. These new tables allow improving compression performances of JPEG algorithm if compared with standard tables. Future research will be devoted to extend such methodology to Huffman encoding of video compression standard (e.g MPEG-2 [9]).

Also the possibility to derive a reliable model able to fit the distribution of the collected events will be considered.

References

1. Battiato, S., Mancuso, M., Bosco, A., Guarnera, M.: Psychovisual and Statistical Optimization of Quantization Tables for DCT Compression Engines, In IEEE Proceedings of International Conference on Image Analysis and Processing ICIAP (2001) 602-606
2. Bauermann, I., Steinbach, E.: Further lossless compression of Jpeg Images. In Proceedings of PCS 2004 – Picture Coding Symposium, CA - USA, (2004)
3. Bookstein, A., Klein, S.T., Raita, T.: Is Huffman Coding Dead?, Computing, Vol.50 (1993) 279-296
4. Epperson, J.F.: An Introduction to Numerical Methods and Analysis, Wiley, 2001
5. ITU-CCITT Recommendation T.81 Information technology – Digital compression and coding of continuous-tone still images – Requirements and Guidelines (1992) | ISO/IEC 10918-1 (1992)
6. Lakhani, G.: Modified JPEG Huffman Coding. In IEEE Transactions on Image Processing, Vol.12 no.2 (2003) 159-169
7. McIntyre, D. R., Pechura, M.A.: Data Compression Using Static Huffman Code-Decode Tables. Commun. ACM, Vol.28 (1985) 612-616

8. Pennebaker, W.B., Mitchell, J.L.: JPEG - Still Image Compression Standard, Van Nostrand Reinhold, NY, (1993)
9. Pillai, L.: Huffman Coding, Xilinx, XAPP616 (v1.0) (2003)
10. Skretting, K., Husøy, J.H., Aase, S.O.: Improved Huffman Coding Using Recursive Splitting, In Proceedings of Norwegian Signal Processing, NORSIG 1999, Symp., Asker, Norway (1999)
11. Wallace, G.K.: The JPEG still picture compression standard, Commun. ACM, vol.34 (1991) 30-44

3D Surface Reconstruction from Scattered Data Using Moving Least Square Method

Soon-Jeong Ahn[1], Jaechil Yoo[2], Byung-Gook Lee[3], and Joon-Jae Lee[3]

[1] Research Center for Advanced Science and Technology, Dongseo University, Busan, 617-716, South Korea
[2] Department of Mathematics, Dongeui University, Busan, 995, South Korea
[3] Department of Computer Engineering, Dongseo University, Busan, 617-716, South Korea
{sjahn, lbg, jjlee}@dongseo.ac.kr, yoo@deu.ac.kr

Abstract. This paper presents an efficient implementation of moving least square(MLS) approximation for 3D surface reconstruction. The smoothness of the MLS is mainly determined by the weight function where its support greatly affects the accuracy as well as the computational time in the mixed dense and scattered data. In a point-set, possibly acquired from a 3D scanning device, it is important to determine the support of the weight function adaptively depending on the distribution and shape of the given scatter data. Particulary in case of face data including the very smooth parts, detail parts and some missing parts of hair due to low reflectance, preserving some details while filling the missing parts smoothly is needed. Therefore we present a fast algorithm to estimate the support parameter adaptively by a raster scan method from the quantized integer array of the given data. Some experimental results show that it guarantees the high accuracy and works to fill the missing parts very well.

1 Introduction to MLS Approximation

There are many methods to reconstruct the continuous 3D surface from discrete scattered data. The moving least square(MLS) method is introduced to interpolate the irregularly spaced data. The relationship between MLS and G.Backus and F. Gilbert [2] theory was found by Abramovici [1] for Shepard's method and for the general case by Bos and Salkauskas [3]. For scattered data $X = \{x_i\}_{i=1}^n$ in $\mathbb{R}^d$ and data values $\{f(x_i)\}_{i=1}^n$, the MLS approximation of order m at a point $x \in \Omega \subset \mathbb{R}^d$ is the value $p^* \in \Pi_m$ is minimizing, among all $p \in \Pi_m$, the weighted least-square error

$$\sum_{i=1}^{n} (p(x_i) - f(x_i))^2 \theta(||x - x_i||), \tag{1}$$

where θ is a non-negative weight function and $||\cdot||$ is the Euclidian distance in $\mathbb{R}^d$. To get the local approximation $\theta(r)$ must be fast decreasing as $r \to \infty$. So D. Levin [4] suggested the weight function for approximation

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 719–726, 2005.

$$\eta(||x_i - x||) = \exp(\frac{||x_i - x||^2}{h^2}) \tag{2}$$

and prove that (1) is equivalent to find the coefficient vector $\bar{a} = \{a_1, a_2, \cdots, a_n\}$ for approximation $\hat{f}(x) = \sum_{i=1}^{I} a_i f(x_i)$ by minimizing the quadratic form

$$Q = \sum_{i=1}^{n} \eta(||x_i - x||)a_i^2, \tag{3}$$

subject to the linear constraints

$$\sum_{i=1}^{n} a_i p_j(x_i) = p_j(x), \quad j = 1, \cdots, J = \begin{pmatrix} d+m \\ m \end{pmatrix} \tag{4}$$

with $\eta(||x_i - x||) = \theta(||x_i - x||)^{-1}$ in (1). The coefficient vector $\bar{a}$ is determined by

$$\bar{a} = D^{-1}E(E^t D^{-1} E)^{-1}\bar{c}, \tag{5}$$

where $D = 2\text{Diag}\{\eta(||x_1 - x||), \cdots, \eta(||x_n - x||)\}$, $E_{ij} = p_j(x_i)$ and $\bar{c} = (p_1(x), \cdots, p_J(x))^t$. Here h is an average distance between the data points and $p_j, j = 1, \cdots, J$ are the fundamental polynomials in Π_m. This uses the whole given data, so that it takes too long time to process with the large numbers of data.

D. Levin and H.Wendland [5] suggested the parameter s to choose local data and keep the high approximation order but it is very slow and fit to real data because of the fixed

$$h = \sup_{x \in \Omega} \min_{x_j \in X} ||x - x_j||, \tag{6}$$

so that we suggest the new algorithm to determine the local parameter s and take h adaptively according to data in section 2. By calculating RMS error for 5 smooth test functions and comparing the results with Levin method, we demonstrate the accuracy of our method. The results show that the adaptiveness of h is very important and it works well for filling hole case in section 3.

2 A Local and Fast Algorithm

2.1 Fixed h Local MLS Algorithm

To generate C^2 surface, we let $m = 3$ and $d = 2$ which mean the degree of approximant and the dimension of domain, respectively. Let $P = \{(x_i, y_i, z_i) \in \mathbb{R}^3\}_{i=1}^n$ be the set of scattered sample data and $\Omega = \Omega_x \times \Omega_y$ be the rectangular domain which contains the given a set of its projected points to xy-plane. By dividing $\Omega_x = [a, b]$ and $\Omega_y = [c, d]$ uniformly with respect to any fixed resolution(n_x, n_y), we get the evaluation points $W = \{\omega_{ij} \in \mathbb{R}^2 | 0 \le i \le n_x, 0 \le j \le n_y\}$, where $d_x = (b - a)/n_x$, $d_y = (d - c)/n_y$. To determine h, we do not calculate the distance between evaluation point and data points unlike the previous methods.

This is achieved by mapping sample data to grid point and calculating their distribution using simple raster scan method. When the sample data is mapped into grid point, one more points can be mapped into the same grid point. In this case we take their average value for each x, y, z component as representative and index it. For each $\omega \in W$, we can find the numbers of grids from ω_{ij} to nonzero grid point along right, left, up and down direction, denoting it by $q_\omega^r, q_\omega^l, q_\omega^u, q_\omega^d$, respectively. And then we set $q_\omega = \frac{1}{4}(q_\omega^r + q_\omega^l + q_\omega^u + q_\omega^d)$. By taking $q = \max_{\omega \in W}(q_\omega)$, we can set

$$h = q \cdot \max(d_x, d_y) \tag{7}$$

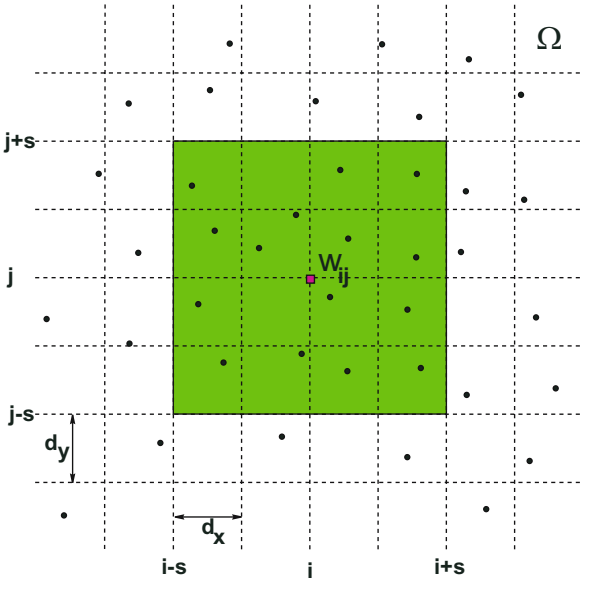


Fig. 1. Domain Ω and local extracted sample data domain

Next, sample data is chosen locally within window of size $2s \times 2s$, where

$$s = [\frac{3}{\sqrt{2}} \cdot q], \tag{8}$$

centered at each evaluation point ω and [ ] denotes the Gauss's symbol. This comes from the property of the weight function related to Gaussian function and the relation between h and q. Actually the standard deviation σ in Gaussian function is related to h, like $2\sigma^2 = h^2$. Because data within 6σ is contributed to the accuracy of the approximated function, we can get the above equation (8). Under these initial conditions, we extract the local subdata set consisted with the above representative data. Since the wanted degree of approximated

function is 3, we must increase the magnitude of s by one until the number of data in subdata set is greater than 10. Using this the final local subdata, we calculate the coefficient vector.

2.2 Adaptive h Local MLS for Data with Holes

If the sample data have big holes, the above fixed h algorithm is unsuitable to reconstruct the original image as shown in Fig. 4 (b),(c). Larger the size of holes is, bigger h is taken. Then the dense part, like nose, eyes and lips, are getting to lose their characteristic property. Therefore, we do not use global q but use local q_ω for each $\omega \in W$. Then the adaptive h_ω

$$h_\omega = q_\omega \cdot \max(d_x, d_y) \tag{9}$$

and the initial adaptive s_ω

$$s_\omega = [\frac{3}{\sqrt{2}} \cdot q_\omega]. \tag{10}$$

Under the adaptive initial conditions, we follow the same process as fixed algorithm. Refer the experimental results in section 3.4.

3 Experimental Results and Conclusions

3.1 Reconstruction Accuracy

In this section we demonstrate the accuracy of our proposed algorithm by the use of 5 test function as follows:

$$\begin{aligned}
g_1(x,y) &= 0.75\exp[-((9x-2)^2+(9y-2)^2)/4] + 0.5\exp[-((9x-7)^2+(9y-3)^2)/4] \\
&\quad + 0.75\exp[-(9x-2)^2/49 - (9y-2)^2/10] - 0.2\exp[-(9x-4)^2-(9y-2)^2] \\
g_2(x,y) &= (\tanh(9-9x-9y)+1)/9 \\
g_3(x,y) &= (1.25+\cos(5.4y))/(6+6(3x-1)^2) \\
g_4(x,y) &= (\exp[(-81/4)((x-0.5)^2+(y-0.5)^2)])/3 \\
g_5(x,y) &= (\sqrt{64-81((x-0.5)^2+(y-0.5)^2)}/9 - 0.5
\end{aligned}$$

where x, y are in the domain $[0, 1]$. We perform experiments with some sample data generated randomly in $[0, 1]$. Here M100 and M500 means 100 and 500 sample data with 49 and 625 uniform data respectively, while others are randomly sampled. On the other hand $R500$ is 500 totally random data. Fig. 2 is the feature of g_3 and its approximated image. The points on the surface are the sample data for M100 case. For each test function g_i, we can find the accuracy of the approximation f by comparing the normalized RMS(root mean square) error which is divided the RMS error by the difference if maximum and minimum values of g_i between the function values on a dense grid. That is,

$$\text{RMS} = \sqrt{\frac{\sum_{i=0}^{n_x}\sum_{j=0}^{n_y}(g_i(x_i,y_i)-f(x_i,y_j))^2}{(n_x+1)(n_y+1)}},$$

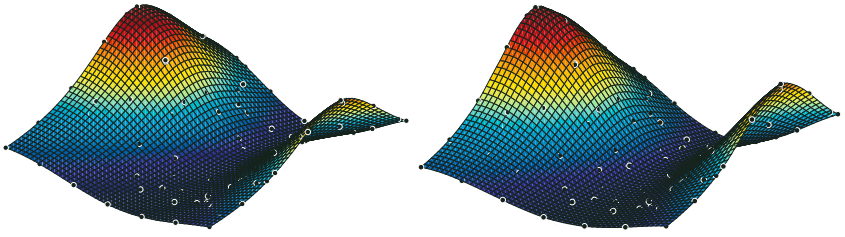


Fig. 2. Feature of g_3(left) and its approximated image for M100(right)

Table 1. Comparison of RMS error between WD and LD for M100, M500 and R500

M100	g_1	g_2	g_3	g_4	g_5
WD	.01076	.00694	.00070	.00199	.00020
LD	.01085	.00699	.00070	.00201	.00020

M500	g_1	g_2	g_3	g_4	g_5
WD	.00079	.00047	.00004	.00013	.00001
LD	.00089	.00045	.00005	.00018	.00001

R500	g_1	g_2	g_3	g_4	g_5
WD	.00080	.00143	.00009	.00020	.00002
LD	.00089	.00152	.00009	.00022	.00004

where $x_i = i/n_x,\ y = j/n_y$ and $n_x = n_y = 50$. Under the same h, we compare the RMS values for each case. Table 1 shows that the fact that MLS approximation theory does not need to have the whole data(WD) but is enough only local data(LD).

3.2 Time Complexity

Table 2 shows that our proposed algorithms makes the processing time faster about 5 times than the use of WD and it is more efficient if the distribution of sample data is more dense and its number is larger.

Table 2. Time complexity between whole data and local data

Time (sec)	M100	M500
WD	10.64	46.22
LD	7.32	9.42

3.3 Filling Holes with Adaptive h Local MLS

Firstly we have experimented for g with additive random noise data of magnitude 20 and some data in original data set is removed to have one hole of the cross shape.

$$g(x,y) = \begin{cases} 50((\sin 2\pi x) + \sin(2\pi y)) & \text{if } x < y, \\ 32((\cos 2\pi x) - \sin(2\pi y)) & \text{o.w.} \end{cases}$$

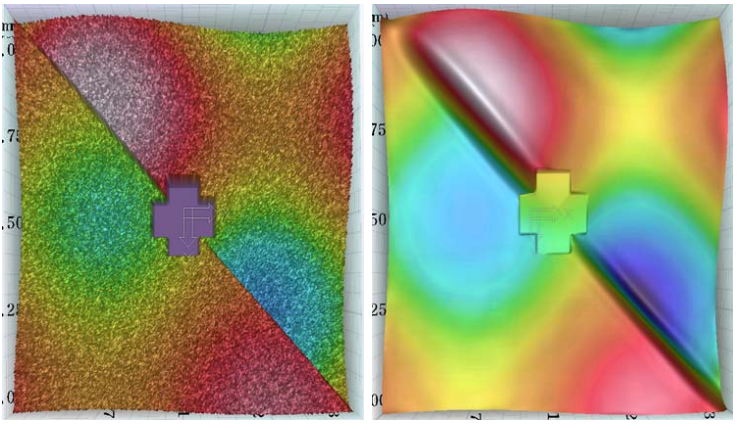


Fig. 3. g function with cross hole by generating noise(left) and its filling hole image(right)

Although the given surface is smooth, if it has missing data with the big sized hole, then fixed h local MLS algorithm is unsuitable. Next, we experiment with real face range data having many noisy data and very big holes. If we take the large fixed h from the largest hole, it can work well for filling holes but it cannot preserve the details. but if we use the adaptive h local algorithm, then we get the nice result like Fig. 4 (d).

3.4 Conclusions and Further Studies

Some experimental results announce us that the proposed algorithm is very simple and efficient with reasonable accuracy for real range data approximation. By introducing h adaptively in every evaluation points, we get the smooth surface that preserves the detail parts such as nose, eyes and lips and fills holes nicely. However, this algorithm occurs some discontinuity on the boundary of each hole due to abrupt change of h near it. So we are researching about multilevel method for getting more smooth results. Some experimental results give us clues that it is very reasonable guess.

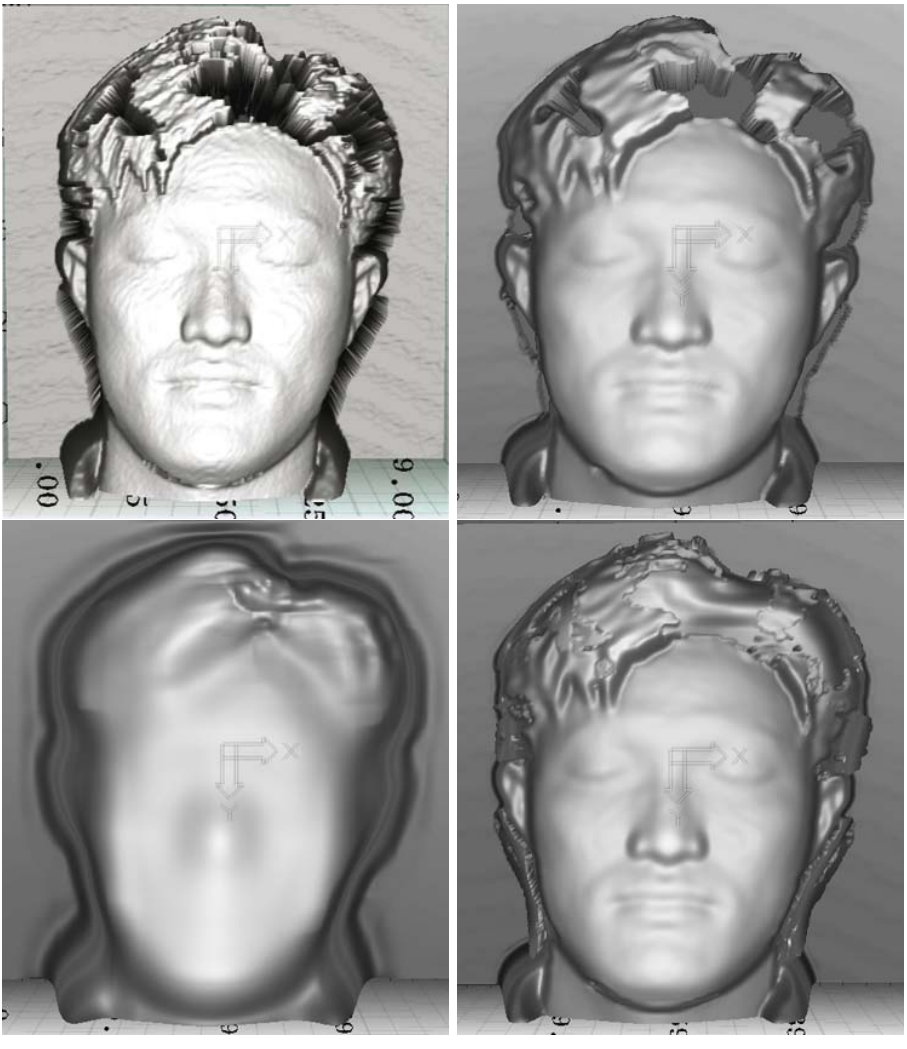


Fig. 4. (a) original face range image, (b) approximated image with fixed $h = 3$, (c) $h = 15$ (d) approximated image with adaptive h

Acknowledgements

1. This work was supported by grant No.R01-2004-000-10851-0,R05-2004-000-10968-0 from Ministry of Science and Technology
2. This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce Industry and Energy of the Korea Government

References

1. F. Abramovici, The Shepard interpolation as the best average of a set of data, Technical Report, Tel-Aviv University, 1984.
2. G. Backus and F. Gilbert, The resolving power of gross earth data, Geophys.J.R. Astr. Soc. **16** 169-205, 1968.
3. L. Bos and K. Salkauskas, Moving least-squares are Backus-Gilbert optimal, J. Approx. Theory **59**(3) 267-275. 1989.
4. D. Levin, The approximation power of moving least-squares, Mathematics of computation, vol 67, **224**, 1517-1531, 1998.
5. Wendland, H., Local polynomial reproduction and moving least squares approxiamtion. IMA J. Numer. Anal. **21**, 285-300. 2001.

A Novel Genetic Programming Based Approach for Classification Problems

L.P. Cordella[1], C. De Stefano[2], F. Fontanella[1], and A. Marcelli[3]

[1] Dipartimento di Informatica e Sistemistica,
Università di Napoli Federico II,
Via Claudio, 21 80125 Napoli – Italy
{cordel, frfontan}@unina.it

[2] Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, Università di Cassino,
Via G. Di Biasio, 43 02043 Cassino (FR) – Italy
destefano@unicas.it

[3] Dipartimento di Ingegneria dell'Informazione e Ingegneria Elettrica,
Università di Salerno,
84084 Fisciano (SA) – Italy
amarcelli@unisa.it

Abstract. A new genetic programming based approach to classification problems is proposed. Differently from other approaches, the number of prototypes in the classifier is not a priori fixed, but automatically found by the system. In fact, in many problems a single class may contain a variable number of subclasses. Hence, a single prototype, may be inadequate to represent all the members of the class. The devised approach has been tested on several problems and the results compared with those obtained by a different genetic programming based approach recently proposed in the literature.

1 Introduction

In the last years several modern computational techniques have been introduced for developing new classifiers [1]. Among others, evolutionary computation techniques have been also employed. In this field, genetic algorithms [2] and genetic programming [3] have mostly been used. The former approach encodes a set of classification rules as a sequence of bit strings. In the latter approach instead, such rules, or even classification functions, can be learned. The technique of Genetic Programming (GP) was introduced by Koza [3] in 1987 and has been applied to several problems like symbolic regression, robot control programming, classification, etc. GP based methodologies have demonstrated to be able to discover underlying data relationships and to represent these relationships by expressions.

GP has already been successfully used in many different applications [4,5]. Although genetic algorithms have often been used for dealing with classification problems, only recently some attempts have been made to solve such problems

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 727–734, 2005.

using GP [6,7,8]. In [7], GP has been used to evolve equations (encoded as derivation trees) involving simple arithmetic operators and feature variables, for hyper-spectral image classification. In [6], GP has also been employed for image classification problems. In [8], an interesting method which considers a c-class problem as a set of c two-class problems has been introduced. In all the above quoted approaches, the number c of classes to be dealt with is used to divide the data set at hand in exactly c clusters. Thus, these approaches do not take into account the existence of subclasses within one or more of the classes in the analyzed data set.

We present a new GP based method for determining the prototypes in a c-class problem. In the devised approach, the prototypes describing samples belonging to c different classes, with $c \geq 2$, consist of logical expressions. Each prototype is representative of a cluster of samples in the training set and consists of a set of assertions (i.e. logical predicates) connected by Boolean operators. Each assertion establishes a condition on the value of a particular feature of the samples in the data set to be analyzed. The number of expressions is variable and may be greater or equal to the number of classes of the problem at hand. In fact, in many classification problems a single class may contain a variable number of subclasses. Hence, c expressions may not be able to effectively classify all the samples, since a single expression might be inadequate to express the characteristics of all the subclasses present in a class. The devised approach, instead, is able to automatically finding all the subclasses present in the data set, since a class is encoded by a variable number of logical expressions. The length of a single expression, i.e. the number of predicates contained in it, is also variable. Each expression may represent either a class or a subclass of the problem. The proposed method works according to the evolutionary computation paradigm. The set of prototypes describing all the classes make up an individual of the evolving population. Given an individual and a sample, classification consists in attributing the sample to one of the classes (i.e. in associating the sample to one of the prototypes). The recognition rate obtained on the training set when using an individual, is assigned as fitness value to that individual. At any step of the evolution process, individuals are selected according to their fitness value. At the end of the process, the best individual obtained, constitutes the set of prototypes to be used for the considered application. Our method for automatic prototyping has been tested on three publicly available databases and the classification results have been compared with those obtained by another GP based approach [9]. In this method individuals are also represented by trees, but expressions involve simple arithmetic operators and constants. Differently from our approach, the number of expressions making up an individual is a priori fixed.

2 Description of the Approach

In our approach a set of prototypes, each characterizing a different class or subclass, consists of a set of logical expressions. Each expression may contain a variable number of predicates holding for the samples belonging to one class in

the training set taken into account. A predicate establishes a condition on the value of a particular feature. If all the predicates of an expression are satisfied by the values in the feature vector describing a sample, we say that the expression *matches* the sample. Training the classifier is accomplished by means of the evolutionary computation paradigm described in Section 3 and provides a set of labeled expressions (i.e. of labeled prototypes). Note that different expressions may have the same label in case they represent subclasses of a class. Given a data set and a set of labeled expressions, the classification task is performed in the following way: each sample of the data set is matched against the set of expressions and *assigned* to one of them (i.e. to a subclasses) or rejected. Different cases may occur:

1. The sample is matched by just one expression: it is assigned to that expression.
2. The sample is matched by more than one expression with different number of predicates: it is assigned to the expression with the smallest number of predicates.
3. The sample is matched by more than one expression with the same number of predicates and different labels: the sample is rejected.
4. The sample is matched by no expression: the sample is rejected.
5. The sample is matched by more than one expression with equal label: the sample is assigned to the class the expressions belong to.

Hereinafter, this process will be referred to as *assignment* process, and the set of samples assigned to the same expression will be referred to as *cluster*.

3 Learning Classification Rules

As mentioned in the introduction, the prototypes to be used for classification are given in terms of logical expressions. Since logical expressions may be thought of as computer programs, a natural way for introducing them in our learning system is that of adopting the GP paradigm. Such paradigm combines genetic algorithms and programming languages in order to evolve computer programs of different complexity for a given task. According to this paradigm, populations of computer programs are evolved by using the Darwin's principle that evolution by natural selection occurs when the replicating entities in the population possess the *heritability* characteristic and are subject to *genetic variation* and *struggle to survive* [2].

Typically, GP starts with an initial population of randomly generated programs composed of functionals and terminals especially tailored to deal with the problem at hand. The performance of each program in the population is measured by means of a *fitness* function, whose form also depends on the problem faced. After the fitness of each program has been evaluated, a new population is generated by selection, recombination and mutation of the current programs, and replaces the old one. This process is repeated until a termination criterion is satisfied. In order to implement such paradigm, the following steps have to be executed:

Table 1. The grammar for the random program generator. N is the dimension of the feature space. Nonterminal symbols are denoted by capital letters.

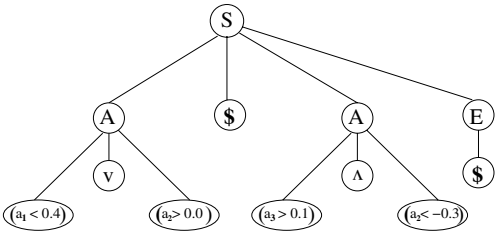
Rule number	Rule	Probability
1	$S \longrightarrow A\$AE$	1.0
2	$E \longrightarrow A\$E\|\$$	0.2, 0.8
3	$A \longrightarrow ABA\|C$	0.2, 0.8
4	$B \longrightarrow \vee\|\wedge$	equiprobable
5	$D \longrightarrow (P > V)\|(P < V)$	equiprobable
6	$P \longrightarrow a_0\|a_1\|\ldots\|a_N$	equiprobable
7	$V \longrightarrow +0.XX\| - 0.XX$	equiprobable
8	$X \longrightarrow 0\|1\|2\|3\|4\|5\|6\|7\|8\|9$	equiprobable

– definition of the structure to be evolved;
– choice of the fitness function;
– choice of the selection mechanism and definition of the genetic operators;

3.1 Structure Definition

The implementation requires a program generator, providing syntactically correct programs, and an interpreter for executing them. The program generator is based on a grammar written for S-expressions. A grammar $\mathcal{G}$ is defined as a quadruple $\mathcal{G} = (\mathcal{T}, \mathcal{N}, S, \mathcal{P})$, where $\mathcal{T}$ and $\mathcal{N}$ are disjoint finite alphabets. $\mathcal{T}$ is said the *terminal alphabet*, whereas $\mathcal{N}$ is said the *nonterminal alphabet*. S, is *the starting symbol* and $\mathcal{P}$ is the set of *production rules* used to define the strings belonging to the language, usually indicated by $v \longrightarrow w$ where v is a string on $(\mathcal{N} \cup \mathcal{T})$ containing at least one nonterminal symbol, and w is an element of $(\mathcal{N} \cup \mathcal{T})^*$. The grammar employed is given in Table 1.

Each individual in the initial population is generated by starting with the symbol S that, according to the related production rule can be replaced only by the string "A$AE". The symbol A can be replaced by any recursive combination of logical predicates whose arguments are the occurrences of the elements in the feature vector. It is worth noting that the grammar has been defined so as to generate individuals containing at least two logical expressions. The role of the nonterminal symbol E and the corresponding production rule is that of adding new expressions to an individual. The terminal symbol $ has been introduced to delimit different logical expressions within an individual. Summarizing, each individual is seen as a unique derivation tree where the leaves are the terminal symbols of the grammar that has been defined for constructing the set of logical expressions to be used as prototypes. Usually, in the literature, in analogy with the phenomena of the natural evolution, a derivation tree is denoted as *genotype* or *chromosome*. Visiting a derivation tree in depth first order and copying into a string the symbols contained in the leaves, we obtain the desired set of logical expressions separated by the symbol $. This string is usually called *phenotype*. Since the grammar is non-deterministic, to reduce the probability of generating too long expressions (i.e. too deep trees) the action carried out by a production



$$(a_1 < 0.4) \vee (a_2 > 0.0)\$(a_3 > 0.1) \wedge (a_2 < -0.3)\$$$

Fig. 1. Example of an individual consisting of two expressions: its tree (the genotype or chromosome) and the corresponding string (the phenotype).

rule is chosen on the basis of fixed probability values (see Table 1). Moreover, an upper limit has been imposed on the total number of nodes contained in a tree. An example of chromosome of an individual is shown in Fig. 1.

The interpreter is implemented by an automaton which computes Boolean functions. Such an automaton accepts in input an expression and a sample and returns as output the value true or false depending on the fact that the expression matches or not the sample.

3.2 Training and Fitness Function

The system is trained with a set containing N_{tr} samples. The training set is used for evaluating the fitness of the individuals in the population. This process implies the following steps:

1. The assignment of the training set samples to the expressions belonging to each individual is performed. After this step, p_i ($p_i \geq 0$) samples have been assigned to the i-th expression. Note that each expression for which $p_i > 0$ is associated with a cluster. In the following these expressions will be referred to as *valid*. The expressions for which $p_i = 0$ will be ignored in the following steps.
2. Each valid expression of an individual is labeled with the label most widely represented in the corresponding cluster.
3. For every individual (i.e. a set of prototypes) a classifier is built up and its recognition rate is evaluated. Such rate is assigned as fitness value to the individual.

In order to favor those individuals able to obtain good performances with a lesser number of expressions, the fitness of each individual is increased by $0.1/N_c$, where N_c is the number of expressions in an individual.

3.3 Selection Mechanism and Genetic Operators

The selection mechanism is responsible for choosing, in the current population, the individuals that will undergo genetic manipulation for producing the new

population. The tournament method has been chosen as selection mechanism in order to control loss of diversity selection intensity [10].

As previously seen in Section 3.1, the individuals are encoded as derivation trees and represent the chromosomes to which the genetic operators are applied. This encoding allows to implement the actions performed by the genetic operators as simple operations on the trees. The individuals in the population are modified using two operators: *crossover* and *mutation*. Both these operators preserve the syntactic correctness of the expressions making up the new individuals generated.

The crossover operator works with two chromosomes C_1 and C_2 and yields two new chromosomes. These new chromosomes are obtained by swapping parts of the trees (i.e. subtrees) of the initial chromosomes C_1 and C_2. The crossover operates by randomly selecting a nonterminal node in the chromosome C_1 and a node of C_2 with the same nonterminal symbol. Then, it swaps the derivation subtrees rooted under the selected nodes. From a phenotype perspective, the result of applying the crossover is swapping substrings representing parts of two chromosomes. The sizes of the swapped parts depend on the nonterminal symbol chosen. For instance, with reference to Table 1, if the symbol chosen is A, then the swapped substrings contain at least an entire predicate or even an entire expression. On the contrary, if the symbol chosen is X, then only single digits of the two strings are swapped.

Given a chromosome C, the mutation operator is applied by randomly choosing a nonterminal node in C and then activating the corresponding production rule in order to substitute the subtree rooted under the chosen node. The effect of the mutation depends on the nonterminal symbol chosen. In fact, this operation can result either in the substitution of the related subtree, causing a macro-mutation, or in a simple substitution of a leaf node (micro-mutation). For instance, considering the grammar of Table 1, if a node containing the symbol D is chosen, then the whole corresponding subtree is substituted. In the phenotype, this operation causes the substitution of the predicates encoded by the old subtree with those encoded by the new generated subtree. If, instead, the symbol B is chosen, only a leaf of the tree is substituted, causing, in the phenotype, the substitution of a boolean operator with one of those in the right side of the rule.

4 Experimental Results

Three publicly available data sets [11] have been used for training and testing the previously described approach (see Table 2). The IRIS data set is very fa-

Table 2. The class distribution is shown within the parentheses of the last column

Name	Classes	Features	Size
IRIS	3	4	150(50+50+50)
BUPA	2	6	345(145+200)
Vehicle	4	18	846(212+217+218+199)

mous in the literature and consists of 150 samples of iris flowers belonging to three different classes equally distributed in the set. BUPA is a Liver Disorders data set, while the Vehicle data set is made of feature vectors describing vehicle images. In order to use the grammar shown in Table 1 the features of the data sets taken into account have been normalized in the range $[-1.0, 1.0]$. Given a not normalized sample $\mathbf{x} = (x_1, \ldots, x_N)$, every feature x_i is normalized using the formula:

$$x_i = \frac{x_i - \overline{x}_i}{2\sigma_i}$$

where $\overline{x}_i$ and σ_i, respectively represent the mean and the standard deviation of the i-th feature computed over the whole data set.

Each data set has been divided in two parts, the first one used as training set and employed in the fitness evaluation of the individuals, the second one used as test set in order to evaluate the classifier performance (i.e. the recognition rate) at the end of the evolution process. The two data sets are disjoint and randomly generated from the original data sets. The values of the evolutionary parameters, used in all the performed experiments, have been heuristically determined and are: Population size = 200; Tournament size = 10; Elithism size = 5; Crossover probability = 0.4; Mutation probability = 0.8; Number of Generations = 300; Maximum number of nodes in an individual = 1000.

The performance of the proposed classification scheme has been evaluated by averaging the recognition rate over 10 runs, using a 3-fold cross validation procedure. The data set has been divided into three parts alternatively used as test set. 10 runs have been performed with different initial population, but keeping unchanged all the other parameters. Hence, 30 runs have been performed for each data set. In Table 3 the results obtained by our method are shown in comparison with those obtained by the GP based approach presented in [9], in which the number of clusters to be found is a priori fixed and set equal to the number of classes of the problem at hand. Since the GP approach is a stochastic algorithm, the standard deviations are also shown. Moreover, the average numbers of found clusters (represented by valid expressions in the considered individual) and the related standard deviations are reported. The experimental results show that the proposed method obtains higher recognition rates than the method used for comparison, on all the data sets.

Table 3. The average recognition rates (%) for the compared classifiers and the average number of data clusters found by the system

Data sets	other GP	our GP	S.D.	N_c	σ_{N_c}
IRIS	98.67	99.4	0.5	3.03	0.2
BUPA	69.87	73.8	3.0	2.36	0.5
Vehicle	61.75	65.5	2.0	4.8	0.6

5 Conclusions

A new genetic programming based approach to classification problems has been proposed. The prototypes of the classes consist of logical expressions establishing conditions on feature values and thus describing clusters of data samples. The proposed method is able to automatically discover the clusters contained in the data, without forcing the system to find a predefined number of clusters. This means that a class is not necessarily represented by one single prototype or by a fixed number of prototypes. On the contrary, other methods, namely the one used for comparison, a priori set the number of possible clusters. The greater flexibility of our method depends on the dynamic labeling mechanism of logical expressions. As already mentioned, the labels of the expressions of each individual in the new population generated at every iteration, are assigned so as to maximize the recognition rate of the classifier based on that individual. The experimental results obtained on three publicly available data sets show a significant improvement with respect to those reported in the literature.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & sons, Inc. (2001)
2. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press (1992)
3. Koza, J.R.: Genetic programming: On the programming of computers by means of natural selection. Statistics and Computing **4** (1994)
4. Bastian, A.: Identifying fuzzy models utilizing genetic programming. Fuzzy Sets and Systems **113** (2000) 333–350
5. Koppen, M., Nickolay, B.: Genetic programming based texture filtering framework. Pattern recognition in soft computing paradigm (2001) 275–304
6. Agnelli, D., Bollini, A., Lombardi, L.: Image classification: an evolutionary approach. Pattern Recognition Letters **23** (2002) 303–309
7. Rauss, P.J., Daida, J.M., Chaudhary, S.A.: Classification of spectral image using genetic programming. In: GECCO. (2000) 726–733
8. Kishore, J.K., Patnaik, L.M., Mani, V., Agrawal, V.K.: Application of genetic programming for multicategory pattern classification. IEEE Transactions on Evolutionary Computation **4** (2000) 242–258
9. Muni, D.P., Pal, N.R., Das, J.: A novel approach to design classifiers using genetic programming. IEEE Trans. Evolutionary Computation **8** (2004) 183–196
10. Blickle, T., Thiele, L.: A comparison of selection schemes used in genetic algorithms. Technical Report 11, Gloriastrasse 35, 8092 Zurich, Switzerland (1995)
11. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)

Machine Learning on Historic Air Photographs for Mapping Risk of Unexploded Bombs

Stefano Merler, Cesare Furlanello, and Giuseppe Jurman

ITC-irst - Trento, Italy
{merler,furlan,jurman}@itc.it

Abstract. We describe an automatic procedure for building risk maps of unexploded ordnances (UXO) based on historic air photographs. The system is based on a cost-sensitive version of AdaBoost regularized by hard point shaving techniques, and integrated by spatial smoothing. The result is a map of the spatial density of craters, an indicator of UXO risk.

1 Introduction

More than 1 million bombs were air-dropped by the Allied Forces during WWII in Italy, and at least 10% did not explode. At least 25,000 unexploded ordnances are thus likely to remain buried, in the optimistic estimate that 3 out of 4 were found and correctly disposed of. Only in Trentino, we have knowledge of 32,019 high explosive bombs (801 of which with Long Delay fuses) aimed at 271 attack targets, accounting for 800 – 1,280 UXO still to be found. After more than 50 years, the ordnances are still operative: unexploded bombs constitute a risk for population in case of accidental discovery, and a high inconvenience for any intervention (e.g., construction of infrastructures, remediation of areas), often delaying works, obliging also to evacuations of thousands inhabitants and to communication blockades.

During the war, over 30 millions aerial photographs were taken for target identification, damage assessment, mapping, and other purposes and still survive in the archives of Keele (UK) and NARA (US). On the basis of available mission data, reconnaissance imagery may be acquired from archives, digitalized and geocoded. Images are selected in order to cover all locations in the study area, and at different dates in the case of repeated attacks to targets.

In this paper we present a learning methodology for the detection of the areas potentially including unexploded ordnances. The classification problem is in many aspects similar to that of identifying volcanoes on Venus [1]. Our task is characterized by two main characteristics: the requirement of high specificity (to reduce the number of false alarms) and the high number of test points (making unfeasible the use of several computationally intensive algorithms). Moreover, it is worthwhile noting that there is no need to precisely identify all the single craters: the detection of their clusters is the more informative target. The proposed solution, SSTBoostReg, extends the classical AdaBoost [2] algorithm and it accounts for the above described issues by means of a regularized cost-sensitive variant coupled to a spatial smoother.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 735–742, 2005.

2 Data Description and Preprocessing

A set of $1,464$ images, taken at different dates throughout the whole WWII period, and rather heterogeneous, in terms of both general quality and ground resolution, was selected for the described detection task. The general quality of the images is affected by sky conditions (presence of clouds or smoke), exposure hour (light conditions, brightness and contrast), exposure season (snow may cover the craters, foliages may change) and preservation status. The pixel ground resolution - approximately between 0.25×0.25 and 1×1 m^2 - depends on the flight altitude which ranges from $9,000$ to $25,000$ feet because of the enemy artillery opposition.

The images were scanned at 600dpi resolution and geolocated within the Geographic Information System GRASS (`http://grass.itc.it`). The ground resolution of all the images was reduced to 1.5×1.5 m^2 for noise reduction and uniformity purposes. Fifty images chosen from different reconnaissance flights and of different quality were retained for training set extraction, learning and evaluation of classification models, whereas all the remaining photographs were used for building the risk map.

The training set consists of a set $\{(\mathbf{v}_i, y_i)\}_{i=1,\ldots,N}$ of $k \times k$ pixel windows around examples of the patterns one is trying to recognize. Each pixel window $\mathbf{v}_i$ is seen as a vector of $\mathbb{R}^{k^2}$ whose label y_i belongs to $\{-1, 1\}$. Taking into account the pattern variability, we selected $1,639$ positive examples (a total of about $30,000$ craters is expected to be found in all the study area) and $2,367$ negative ones from 30 out of 50 training images (the remaining 20 images being only used for model evaluation), for a total of $N = 4,006$ examples. The window size was set to 19.5×19.5 m^2, corresponding to $k = 13$ pixels. We tried to include all the possible observable patterns into the training set. Examples from the two classes are shown in Fig. 1a and 1b. In particular, it should be observed that the examples of trees appear very similar to those of craters.

It is worthwhile mentioning that the extraction of training data was not trivial, both for the positive and the negative examples, because of the heterogeneity of the involved patterns. All the training examples were normalized with respect to the local brightness to make it independent from the light conditions of the particular reconnaissance flight according to the transformation $\tilde{\mathbf{v}}_i = (\mathbf{v}_i - \mu_i)/\sigma_i$ for $i = 1, \ldots, N$, where μ_i and σ_i represent the mean value and the standard deviation of the training example $\mathbf{v}_i$ respectively.

For feature reduction, a principal component analysis was performed on the positive samples of training set. A graphical representation of the Eigencraters (i.e., the eigenvectors corresponding to the positive samples) in order of non increasing eigenvalues is shown in Fig. 1c. We took into account only the first $n = 11$ principal components, accounting for more than 95% of the total variance, for the development of the classification models. The projection of the original data into the subspace spanned by the first n Eigencraters is obtained as $\mathbf{x}_i = \tilde{\mathbf{v}}_i[\mathbf{e}_1, \ldots, \mathbf{e}_n]$, where $[\mathbf{e}_1, \ldots, \mathbf{e}_n]$ is the matrix of the Eigencraters in order of non increasing eigenvalues. Therefore, the training set becomes now the ensemble $\{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,N}$.

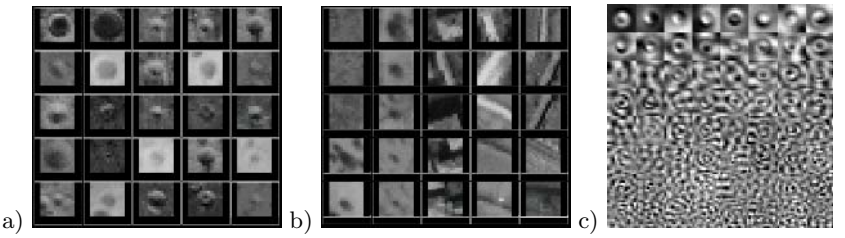


Fig. 1. Examples of positive (left panel) and negative (middle panel) samples (including examples of trees, one of the most confounding patterns, buildings and roads) and Eigencraters reported in order of non increasing eigenvalue (right panel)

3 The SSTBoostReg Model

Standard classification techniques like Nearest Neighbor (NN), Maximum Likelihood Gaussian Classifier (MLGC) and Tree Based Classifiers (TBC) [3] were firstly applied. Bagging [4] and AdaBoost of maximal trees were also considered (100 aggregated models). Performances of these classifiers are reported in Table 1 and they are estimated by 10-fold crossvalidation on the $N = 4,006$ training examples extracted from the 30 out of the 50 training images. AdaBoost achieved the best results in terms of global accuracy (0.92) and performed slightly better than Bagging. However, simpler single classifiers like NN and MLGC achieved higher specificity (0.96) and higher sensitivity (0.89), respectively. Nevertheless, both these last classifiers suffer from not trivial drawbacks. NN classifiers are too slow during the test phase while the computation time is a key factor in our application ($1,464$ test images, corresponding to about 6.10^9 test points); for instance, Bagging and AdaBoost, the fastest models, require an execution time of about 350 hours on a Pentium IV, 3GHz. MLGC is characterized by too low specificity (0.86) and it is very likely to cause many false alarms.

Table 1. Accuracy, sensitivity and specificity (by 10-fold crossvalidation), with the standard deviation, are shown for different standard models and combining methods

Classifier	$Acc. \pm SD$	$Sens. \pm SD$	$Spec. \pm SD$
1NN	0.88 ± 0.02	0.83 ± 0.03	0.93 ± 0.02
3NN	0.90 ± 0.01	0.84 ± 0.02	0.95 ± 0.01
5NN	0.91 ± 0.01	0.84 ± 0.02	0.96 ± 0.01
7NN	0.91 ± 0.01	0.84 ± 0.02	0.96 ± 0.01
MLGC	0.87 ± 0.03	0.89 ± 0.01	0.86 ± 0.02
TBC	0.83 ± 0.02	0.79 ± 0.03	0.85 ± 0.03
Bagging	0.90 ± 0.01	0.85 ± 0.02	0.94 ± 0.01
AdaBoost	0.92 ± 0.01	0.88 ± 0.02	0.95 ± 0.01

Despite the relatively high specificity of the model, the application of AdaBoost to the test images was not satisfactory because of the plethora of false alarms inserted in the test images. On the other side, the percentage of correct recognition of craters resulted to be highly satisfactory. In summary, a specificity of 0.95 is not sufficient to avoid a too high number of false alarms, meanwhile a sensitivity of about 0.75 should be sufficient to identify all the clusters of craters. To achieve higher specificity (at least 0.975), we developed a cost-sensitive variant of AdaBoost, further refined by introducing a regularization factor. A spatial smoothing bivariate function was finally applied.

3.1 The SSTBoost

Cost-sensitive variants of the AdaBoost have been already developed, but they require the introduction of a misclassification cost into the learning procedure [5,6]. Unfortunately, the misclassification costs concerning our task were not available. However, the minimal requirements in terms of accuracy were given as discussed at the beginning of Sec. 3: we could consider satisfactory a model characterized by a very high specificity, at least 0.975, and a sensitivity of at least 0.75. Moreover, there is no need to know the class priors (unknown in our case): the only point is to drive the system into the sensitivity – specificity constraints. To this purpose we have developed SSTBoost (Sensitivity Specificity Tuning Boosting) [7], a variant of AdaBoost where (1) the model error is weighted with separate costs for errors (false negative and false positives) in the two classes, and (2) the weights are updated differently for negatives and positives at each boosting step. Finally, (3) SSTBoost includes a practical search procedure which allows reaching the sensitivity – specificity constraints by automatically selecting the optimal costs. Given a parameter $w \in [0, 2]$, the SSTBoost algorithm allows

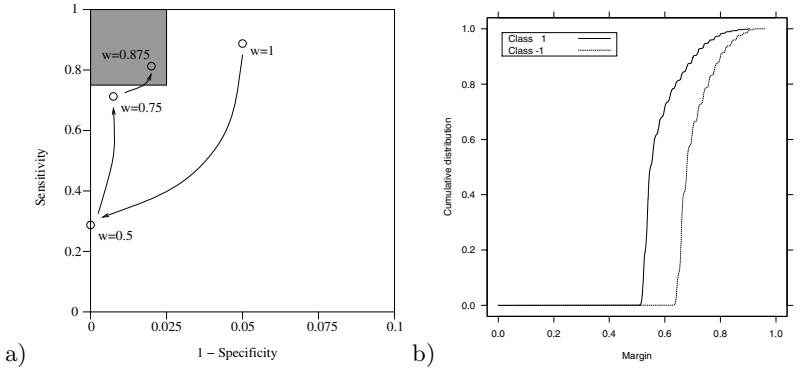


Fig. 2. a) The results of the tuning procedure in terms of sensitivity and specificity, estimated by 10-fold crossvalidation: the parameter w is initialized to 1. The gray square region on top left indicates the constrain target region A. The final value of the parameter is $w^* = 0.875$. b) The corresponding margin distribution for the two classes.

the development of a variant H_w of the AdaBoost characterized by higher sensitivity with respect to the AdaBoost itself when $w > 1$, by higher specificity when $w < 1$. For $w = 1$, we obtain AdaBoost. In [7], a procedure is also proposed for the automatic selection of an optimal cost parameter w^* in order to satisfy or to get as close as possible to admissible sensitivity – specificity constraints. The problem can be addressed as a minimization problem of the real function $\Delta : [0, 2] \rightarrow \mathbb{R}^+$ defined as $\Delta(w) = \text{dist}(\phi_H(w), A) = \min_{a \in A} \|\phi_H(w) - a\|$, describing the distance of a suitable target region A in the ROC space and the ROC curve $\phi_H(w)$. The problem admits a solution, not necessarily unique: the possible optimal cost parameters are selected by $w^* = \text{argmin}_w \Delta(w)$.

We applied the SSTBoost to our training set by using maximal classification trees as base learners (100 models). As discussed in Sec. 3, the target region A was defined by the constraints $Se > 0.75$, $Sp > 0.975$. To obtain predictive estimates of both sensitivity and specificity we used 10-fold crossvalidation. The steps of the automatic procedure for the selection of the parameter w are sketched in Fig. 2a, showing that the procedure falls into the region A in only 4 optimization steps. Fig. 2b can explain how SSTBoost works in terms of the margin. For $w^* = 0.875$ (and in general for $w \neq 1$), the margin of the two classes is differently maximized; the margin of the negative examples results higher than the margin of the positives one. The SSTBoost concentrates more resources to learn the negative examples.

3.2 Regularization

Regularization techniques for AdaBoost have been already discussed in literature, e.g. as in [8]. The alternative method introduced in [9] consists in a bias-variance control procedure based on removal of training data points according to a suitable threshold of an hardness measure, to be interpreted as the classification task difficulty for the predictors. Such hardness measure is linked to the dynamics of the AdaBoost weights, while the threshold can be evaluated by minimizing the generalized error of a loss function. In details, the degree of hardness for each training point $p_i \in D$ can be derived by the misclassification ratio of a point $\rho(p_i)$ in the aggregation of K AdaBoost models M_k, yielding a family of telescopic training subsets $\{D_r\}_{r \in (0,1]} = \{p_i \in D | \rho(p_i) < r\}$, shaved of the r-hardest points. After the optimal parameter is detected, the training set can be pruned by shaving the hardest points and the classifier retrained. We denote this regularized cost-sensitive version of AdaBoost by SSTBoostReg. In the present task, a portion of the training data is put aside as validation set for the threshold estimation: results are discussed in Sec. 4.

3.3 Spatial Analysis

Given a test image, the classification is based on a moving window procedure. The SSTBoostReg is applied and the corresponding output $H_{w^*}(\mathbf{x}) \in [-1, 1]$ is assigned to the central pixel of the window. For a rejection threshold $T_0 = 0$, a crude application of the SSTBoost leads to classify as craters all the pixels such

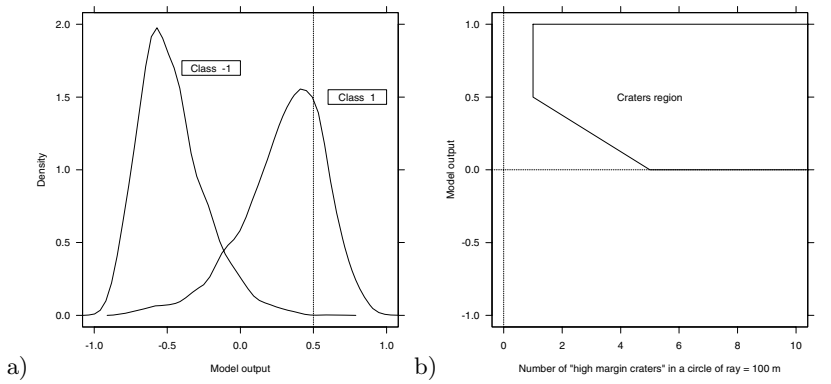


Fig. 3. a) Class densities of the 10-fold SSTBoostReg outputs. The vertical line indicates the threshold T_u for the identification of "high margin craters". b) Class labels are assigned according to the pair model output - density of "high margin craters".

Fig. 4. The left image shows the classification obtained by applying SSTBoostReg, without spatial analysis. The middle image shows the "high margin craters" and the right image includes all the craters recognized by the local thresholding procedure.

that $H_{w^*}(\mathbf{x}) > T_0$ (obviously, groups of adjacent pixels over this threshold shall be aggregated). According to the results reported in Tab. 2, where the different versions of AdaBoost are compared, a percentage of false alarms of about 2% is expected. We now introduce an heuristics for reducing the percentage of false alarms, motivated by the observation that the craters are not randomly spatially distributed but they are clustered near the main targets. As shown in Fig. 3a, nearly ever a negative example is classified as positive with a response above $1/2$ (by $10-$fold crossvalidation). We can thus use a threshold $T_u = 1/2$ for identifying what we call "high margin craters". The idea is now to exploit the spatial density of "high margin craters" for locally modifying the rejection threshold of the model: the higher the density of "high margin craters", the lower the rejection threshold T. In particular, the threshold is decreased linearly from T_u to T_0 as the density of "high margin craters" increase from 1 to 5, and it remains constant for higher densities of "high margin craters" (see Fig. 3b for details). The "high margin craters" identified on a test image are shown in Fig. 4: as can

Table 2. Accuracy, sensitivity and specificity, including standard deviations, for AdaBoost, SSTBoost and SSTBoostReg (all obtained by aggregating 100 tree models)

Classifier	$Acc. \pm SD$	$Sens. \pm SD$	$Spec. \pm SD$
AdaBoost	0.92 ± 0.01	0.88 ± 0.02	0.95 ± 0.01
SSTBoost	0.91 ± 0.01	0.81 ± 0.02	0.98 ± 0.01
SSTBoostReg	0.92 ± 0.01	0.83 ± 0.02	0.98 ± 0.01

be easily seen, no false alarms are introduced at this level. Fig. 4 also includes the craters identified by the local thresholding procedure.

4 Performance Evaluation

The performances of AdaBoost, SSTBoost and SSTBoostReg, computed by 10−fold crossvalidation, are compared in Tab. 2: as expected, SSTBoost outperforms standard AdaBoost in terms of specificity (0.95 for AdaBoost, 0.98 for SSTBoost), meanwhile AdaBoost achieved higher global accuracy (0.92 for AdaBoost, 0.91 for SSTBoost). By looking at the results of the classifiers reported in Tab. 1, only SSTBoost falls into the target region A determined by the sensitivity – specificity constraints. The application of the regularization procedure allowed the global accuracy to be increased, improving from 0.91 of SSTBoost to 0.92 of SSTBoostReg, without decreasing neither specificity (0.98

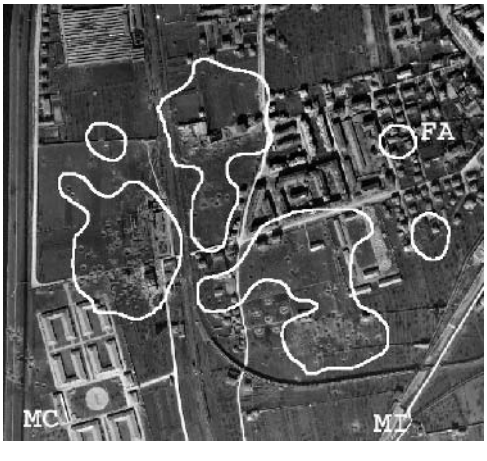


Image	MC	MI	FA
TS1	0	0	1
TS2	3 (13-11-5)	0	1
TS3	1 (9)	2	0
TS4	0	0	1
TS5	0	0	1
TS6	0	0	1
TS7	0	3	1
TS8	0	5	0
TS9	0	0	1
TS10	0	0	2
TS11	1 (16)	0	3
TS12	0	0	1
TS13	0	7	1
TS14	0	5	4
TS15	1 (10)	11	3
TS16	0	2	2
TS17	0	0	4
TS18	1 (18)	6	4
TS19	1 (> 500)	17	1
TS20	0	0	1

Fig. 5. Example of error detection based on the analysis of the risk map: the contour lines indicate regions with non null density of craters (by machine learning). The figure includes examples of missed craters clusters (MC), missed isolated craters (MI) and false alarms (FA). For each test image the values for MC (with relative number of craters), MI and FA are also reported.

for both SSTBoost and SSTBoostReg) nor sensitivity (0.81 for SSTBoost, 0.83 for SSTBoostReg).

The SSTBoostReg procedure coupled with the spatial analysis allowed the avoidance of a high number of false alarms without reducing too much the sensitivity of the model. The final output of the system is a map of the spatial density of craters. Performances were then evaluated in terms of number of false alarms in unbombed areas (FA), number of missed isolated craters (MI) and number of missed clusters of craters (MC), obtained by manual census. An example of error detection is shown in Fig. 5.

A set of 20 images was used for evaluating the model performances. The model failed to identify big clusters of craters only once (but the quality of the corresponding image is really very low). In 5 of the 20 images a small isolated cluster of craters was missed and only a total of 58 isolated craters were missed. The number of false alarms is considerably low, with an average of 1.6 false alarms per image (see Fig. 5).

5 Conclusions

The classification system was developed within the UXB-Trentino Project (`http://uxb.itc.it`), funded by the Province of Trento. The new risk map is used for public consultation in case of building planning, with semi-automatic production of risk reports. Consultation of the risk map under indication of the Civil Defense Department has become a standard procedure; its use as a mandatory procedure for new construction works is being considered.

References

1. Burl, M.C., Asker, L., Smyth, P., Fayyad, U.M., Perona, P., Crumpler, L., Aubele, J.: Learning to recognize volcanoes on Venus. Machine Learning **30** (1998) 165–194
2. Schapire, R., Freund, Y., Bartlett, P., Lee, W.: Boosting the margin: A new explanation for the effectiveness of voting methods. Annals of Statistics **26** (1998) 1651–1686
3. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth, Belmont (CA) (1984)
4. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
5. Ting, K.M., Zheng, Z.: Boosting trees for cost-sensitive classifications. In: European Conference on Machine Learning. (1998) 190–195
6. Fan, W., Stolfo, S.J., Zhang, J., Chan, P.K.: AdaCost: misclassification cost-sensitive boosting. In: Proc. 16th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (1999) 97–105
7. Merler, S., Furlanello, C., Larcher, B., Sboner, A.: Automatic model selection in cost-sensitive boosting. Information Fusion **4** (2003) 3–10
8. Rätsch, G., Onoda, T., Müller, K.: Soft margins for Adaboost. Machine Learning **42** (2001) 287–320
9. Merler, S., Caprile, B., Furlanello, C.: Bias-variance control via hard points shaving. International Journal of Pattern Recognition and Artificial Intelligence **18** (2004) 891–903

Facial Expression Recognition Based on the Belief Theory: Comparison with Different Classifiers

Z. Hammal[1], L. Couvreur[2], A. Caplier[1], and M. Rombaut[1]

[1] Laboratory of images and signals LIS,
46 avenue Félix Viallet, F-38031 Grenoble, France
[2] Signal Processing Laboratory, Faculté Polytechnique de Mons,
1 Avenue Copernic, B-7000, Mons, Belgium

Abstract. This paper presents a system for classifying facial expressions based on a data fusion process relying on the Belief Theory (BeT). Four expressions are considered: *joy, surprise, disgust* as well as *neutral*. The proposed system is able to take into account intrinsic doubt about emotion in the recognition process and to handle the fact that each person has his/her own maximal intensity of displaying a particular facial expression. To demonstrate the suitability of our approach for facial expression classification, we compare it with two other standard approaches: the Bayesian Theory (BaT) and the Hidden Markov Models (HMM). The three classification systems use characteristic distances measuring the deformations of facial skeletons. These skeletons result from a contour segmentation of facial permanent features (mouth, eyes and eyebrows). The performances of the classification systems are tested on the Hammal-Caplier database [1] and it is shown that the BeT classifier outperforms both the BaT and HMM classifiers for the considered application.

1 Introduction

The human-machine interface (HMI) is definitively evolving to an intelligent multi-modal interface, combining various human communication modes. Among others, facial expression is a very efficient mean for human beings to communicate their intention.

In this work, we propose a rule-based system for automatically classifying facial expressions. This system relies on the Belief Theory (BeT). Like other methods [2,3,4], our approach is based on facial deformation features (eyes, eyebrows and mouth). It allows to deal with uncertain data and recognize facial expressions in the presence of intrinsic doubt. Clearly, humans do not behave in a *binary* way: they do not produce *pure* expressions but rather combinations of them. Our system is able to identify either pure expressions as well as mixed ones. In order to demonstrate the efficiency of BeT system for the purpose of facial expression recognition, its performances are compared with those of more classical approaches, namely Bayesian Theory (BaT) and the Hidden Markov Models (HMMs).

Section 2 presents how video data are preliminary processed in order to recognize facial expression. In section 3 we describe the Belief Theory classifier, in section 4 the Bayesian classifier and in section 5 the HMM classifier. Section 6 describes the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 743 – 752, 2005.

video database used in this work and presents a comparison between the performances of the three classifiers.

2 Facial Expression Analysis

In this section, we describe how a video sequence of face images is analysed for recognition of facial expressions. First, the contours of facial features are automatically extracted in every frame using the algorithms described in [5,6] (Fig. 1.a).

Next, five characteristic distances are defined and estimated (Fig. 1.b): eye opening (D_1), distance between the inner corner of the eye and the corresponding corner of the eyebrow (D_2), mouth opening width (D_3), mouth opening height (D_4), distance between a mouth corner and the outer corner of the corresponding eye. These distances form together a characteristic vector associated to each facial expression and can be used for modeling and recognizing facial expressions.

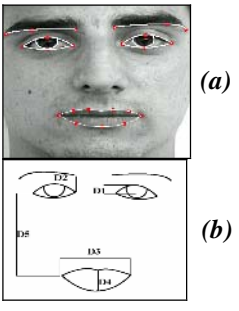


Fig. 1. a) facial features segmentation; b) facial skeleton and characteristic distances.

The BeT recognition process involving all the distances D_i yields to a classification of every frame in the video sequence in terms of a single expression or a mixture of expressions. A state of doubt between two expressions can appear. In order to cope with it, a post processing based on the analysis of transient wrinkles in the nasal root and based on the analysis of the mouth shape is added.

The presence or absence of wrinkles in the nasal root (Fig. 2.a) is detected by using a Canny edge detector. If there is about twice more edges points in the nasal root of the current frame than in the nasal root of a frame with the *neutral* expression, the presence of transient wrinkles is validated, and discarded otherwise. The mouth shape is also used (Fig. 2.b, 2.c). According to the expression, the ratio between length and width of the mouth is larger or smaller than its corresponding value for the *neutral* expression.

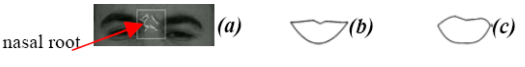


Fig. 2. (a) wrinkles in the nasal root, examples of mouth shapes in case of : (b) *joy*, (c) *disgust*

3 Classification by the Belief Theory

3.1 Definition of the Symbolic States

As shown from our expertise database [1], every characteristic distance D_i can be higher, lower or roughly equal to its value $D_{i,neutral}$ defined for the *neutral* expression, whatever the actual face expression. This comes naturally to the definition of three symbolic states:

- the *higher* state C^+ if D_i is significantly higher than $D_{i,neutral}$;
- the *lower* state C^- if D_i is significantly lower than $D_{i,neutral}$;
- the *neutral* state S if D_i is close to $D_{i,neutral}$.

Hence, one can identify the symbolic state of every characteristic distance for a given face image and analyse its time evolution along a video sequence. Fig. 3 presents the evolution of D_2 (distance between the interior corner of the eye and the interior corner of the eyebrow) and D_5 (distance between one mouth corner and the external corner of the corresponding eye) for several persons and for a given expression. In each case, the video sequence starts with a *neutral* expression, goes towards the actual expression and returns to the neutral expression. Clearly, we observe similar time evolutions of the characteristic distance, thereof the corresponding symbolic state, whatever the subject. This observation also holds for the other characteristic distances and facial expressions.

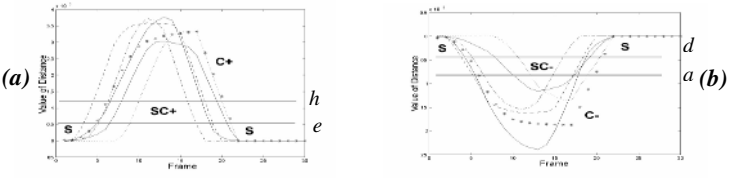


Fig. 3. Examples of time evolution of characteristic distances (a) D_2 and (b) D_5 in case of *surprise* and *joy* face expression, respectively. Thresholds *h*, *e*, *d* et *a* are defined in section 3.3.

3.2 The Belief Theory

Originally introduced by Dempster and Shafer and revisited by Smets [7], the Belief Theory (BeT) can be seen as a generalization of the probability theory. It requires the definition of a set $\Omega=\{E_1, E_2,\ldots,E_N\}$ of N exclusive and exhaustive assumptions. We also consider the power set 2^Ω that denotes the set of all subsets of Ω. To each element A of 2^Ω is associated an elementary piece of evidence *m(A)* which indicates all confidence that one can have in this proposal. The function *m* is defined as:

$$m: \quad 2^\Omega \rightarrow [0,1] \qquad A \mapsto m(A) \quad \text{with} \quad \sum_{A\subseteq\Omega} m(A)=1 \tag{1}$$

In our application, the assumptions E_i correspond to the four facial expressions : *joy* (E_1), *surprise* (E_2), *disgust* (E_3) and *neutral* (E_4); 2^Ω corresponds to single expressions or combinations of expressions, that is, $2^\Omega=\{E_1, E_2, E_3, \ldots, E_1 E_2, E_2 E_3, \ldots\}$, and A is one of its elements.

3.3 Modelling Process

The modelling process aims at computing the state of every distance D_i and at associating a piece of evidence. Let define the basic belief assignment (BBA) m_{Di} as:

$$m_{Di} : \quad 2^{\Omega'} \rightarrow [0,1] \qquad (2)$$
$$B \mapsto m_{Di}(B)$$

With $\Omega' = \{C^+, C^-, S\}$, $2^{\Omega'} = \{C^+, C^-, S, SC^+, SC^-, C^-SC^+\}$, where $S \cup C^+$ (noted SC^+) states the doubt between S and $C+$, $S \cup C^-$ (noted SC^-) states the doubt between S and C^- and $m_{Di}(B)$ is the piece of evidence (PE) of each state B.

A numerical/symbolic conversion is carried out, which associates to each value of Di one of the symbols of $2^{\Omega'}$. To carry out this conversion, we defined a model for each distance using the states of $2^{\Omega'}$ (Fig. 4). We assume that the symbol C^-SC^+ is impossible.

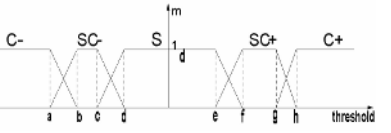


Fig. 4. Model of distances states

In Fig.4., m is the PE associated to each possible state in $2^{\Omega'}$ and the thresholds *(a... h)* are the limit values of D_i corresponding to each state or subset of states. For each distance D_i, the threshold ***h*** (resp. ***a***) of the state C^+ (resp. C^-) corresponds to the average of the maximum (resp. minimal) values of D_i for all the subjects and all the expressions of the expertise database. The thresholds ***d*** and ***e*** of the state S are defined in the same way.

The median of the maximum values of each distance for all the subjects and all the expressions of the expertise database is computed. The thresholds ***f***, ***b*** (resp. ***c, g***) of the intermediate states are defined by mean+median (resp. mean-median) of each state (C^+, C^-, S).

3.4 Recognition Process

Analysis. The analysis of the states for the five distances associated to each of the four expressions (*joy, surprise, disgust* and *neutral*) allows us to exhibit for each expression a specific combination of these states. Table 1 shows the resulting states combinations.

For example, in case of *joy* (E_1), the mouth is opening (C^+ state for D_3 and D_4), the corners of the mouth are going back toward the tears (C^- state for D_5) and the eyebrows are slackened (S state for D_2). The distance between the interior corner of the eye and the interior corner of the eyebrow decreases (C^- state for D_2) and the eyes become slightly closed (C^- state for D_1).

Table 1. Theoretical table of D_i states for each expression

	D_1	D_2	D_3	D_4	D_5
Joy E_1	C^-	S / C^-	C^+	C^+	C^-
Surprise	C^+	C^+	C^-	C^+	C^+
Disgust	C^-	C^-	S / C^+	C^+	S / C^-
Neutral	S	S	S	S	S

The proposed combinations of symbolic states are similar to the MPEG-4 [8] description of the deformations undergone by facial features for such expressions , yet they give some extensions.

Note that in some cases, two different states are possible for a given distance (for example, see D_2 for *joy*, D_3 for *disgust*). This can lead to doubt between two expressions. For example, the classifier is not always able to distinguish *disgust* and *joy* because both expressions can be described by the same combination of states.

The expression E_5 is added as the *unknown* expression or rejection class. It represents all the expressions that do not correspond to any of the state combination in Table 1.

Combination and Decision. We have several sources of information ,namely the distances D_i, to which we associate PEs. Our goal is to obtain a PE which takes into account all the available information. The BBA is obtained using the rule of conjunctive combination or orthogonal sum. In the case of two distances D_1 and D_2, the orthogonal sum is defined in the following way:

$$m = m_{D_1} \oplus m_{D_2} \tag{3}$$

$$m(A) = \sum_{B \cap C = A} m_{D1}(B) m_{D2}(C) \tag{4}$$

A, B and C are expressions or subsets of expression.

This allows to obtain propositions whose number of elements is lower than the initial ones and to associate them a piece of evidence. The final PE is thus more accurate. More explicitly, if one takes two basic belief assignments: $m_{D1}(E_1 \cup E_3)$ $m_{D1}(E_1)$ $m_{D1}(E_2)$

$m_{D2}(E_1)$ $m_{D2}(E_2)$ $m_{D2}(E_1 \cup E_2)$

their combination gives the results of Table 2. The piece of evidence of each expression by the combination of results of the two distances is calculated by:

Table 2. Example of combination of PEs of two distances. ∅ is the empty set

D_1 / D_2	E_1	E_2	$E_1 \cup E_3$
$E_2 \cup E_3$	∅	E_2	E_3
E_1	E_1	∅	E_1
E_2	∅	E_2	∅

$m_{D12}(E_1) = m_{D1}(E_1).\ m_{D2}(E_1) + m_{D1}(E_1)\ m_{D2}(E_1 \cup E_3)$,
$m_{D12}(E_2) = m_{D1}(E_2 \cup E_3).\ m_{D2}(E_2) + m_{D1}(E_2).m_{D2}(E_2)$,
$m_{D12}(E_3) = m_{D1}(E_2 \cup E_3).m_{D2}(E_1 \cup E_3)$,
$m_{D12}(\varnothing) = m_{D1}(E_2 \cup E_3).m_{D2}(E_1) + m_{D1}(E_1).m_{D2}(E_2) + m_{D1}(E_2).m_{D2}(E_1) + m_{D1}(E_2).\ m_{D2}(E_1 \cup E_3)$.

Conflicts, noted ∅, can appear in case of incoherent sources. In the scope of the presented application, the conflict corresponds to a configuration of distance states which does not appear in Table 1. It comes from the fact that Ω is not exhaustive. The additional *unknown* expression or class of reject E_5 represents all these conflicts states (Table 2).

The decision is the ultimate step of the classification process. It consists in making a choice between various assumptions E_i and their possible combinations. Making a choice means taking a risk, except if the result of the combination is perfectly reliable: $m(E_i)=1$. Here, the accepted proposal is the one with maximum value of PE.

4 Bayesian Classifier

In this work, the data and the classes of the Bayesian classifier consist in the distance vectors and the facial expressions, respectively. The statistical models aim at modelling the probability density functions of the observation data for every class.

Here, the probability density functions are defined as mixtures of 3 Gaussian components $N(\mu_k, \Sigma_k)$:

$$p(x|y)=\sum_{k=1}^{3} w_k N(\mu_k, \Sigma_k) \tag{5}$$

where x and y denote the distance vector and the facial expression class, respectively. The parameters of these models, *i.e.* the mean vectors μ_k, the covariance matrices Σ_k and the mixing weights w_k, are estimated in a Maximum Likelihood (ML) sense independently for every class. Since this problem is a missing data problem, it can be addressed by the Expectation-Maximization (EM) algorithm [9] .

During recognition experiments, a distance vector is derived for every frame. Consecutive distance vectors are assumed to be statistically independent as well as the underlying class sequences. This vector is presented to each mixture of Gaussians and its likelihood score is computed. The vector is eventually assigned to the class corresponding to the mixtures of Gaussians with the highest likelihood score:

$$\tilde{y}=\arg\max_{y} p(x|y) \quad y \in \{E_1, E_2, E_3, E_4, E_5\} \tag{6}$$

5 HMM Classifier

Hidden Markov Models (HMM) are widely used in many fields (*e.g.*, automatic speech recognition) [10] where temporal (or spatial) dependencies are present in the data.

For the recognition of facial expressions, we adopt a 5-state HMM, one state per expression, whose topology is depicted in Fig. 5. As can be seen, the HMM forces the state sequence to start in the *neutral* state (E_4), then can stay some times in either the *joy* state (E_1), the *surprise* state (E_2) or the *disgust* state (E_3). An *unknown* state (E_5) is also considered for representing any other expression. Such topology is practically realized by forcing some transition probabilities to zero.

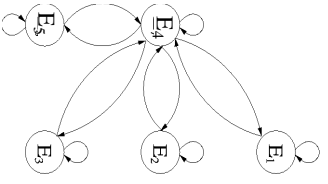


Fig. 5. HMM classifier topology. Branches between the first E_4 and the last E_4 should not be bi-directional. Besides, there should be only one E_4 state to which you loop back after leaving an expression state.

The state probability density functions are defined as mixtures of 3 Gaussian components. All the HMM parameters are estimated by an EM-style algorithm. Given some observation data, *i.e.* sequences of vectors of characteristics distances, and an initial estimate, the HMM parameters are refined using the Baum-Welch algorithm [10]. Note that its Viterbi approximation can be used as well.

During the recognition experiments, a sequence of distance vectors is presented to the HMM. The most likely state sequence is searched by a Viterbi algorithm [10]. Hence, a state is assigned to every distance vector, equivalently an expression is assigned to every frame. Unlike for the Bayesian classifier, the state choice is not taken independently at each time instant but rather globally it is assumed that there

exists some dependency between consecutive face classes. Such criterion allows to recover many errors of the Bayesian classifier.

6 Results and Discussion

6.1 Database

The Hammal-Caplier database is used for our experiments (21 subjects and 4 expressions) [1]. Each video recording starts and ends with a *neutral* state (for example Fig. 6). The sequences have been acquired during 5 seconds at 25 images/second.

Fig. 6. Examples of expressions. Each record starts and finishes with a neutral state

For the expertise step of the BeT, 1170 frames (13 subjects and 4 expressions) of the Hammal-Caplier expertise database have been considered. All the frames of the expertise database are segmented and the five distances defined on Fig. 1 are computed and used in order to define the thresholds of section 3.3 and to build Table 1.

In order to evaluate the robustness to different variations (gender, ethnicity, difference of expressions ,etc), the BeT system is tested on the Hammal-Caplier test database (630 frames for 8 subjects and 4 expressions).

For the HMM and Bayesian classifiers all the data of Hammal-Caplier database are used for the training step and the test is carried out by a 21-fold cross validation. It consists in taking the data from 20 out of 21 subjects for the training step and in using the data of the remaining subject for the test step. This process is repeated 21 times, considering a different test subject each time. The classification rate is the average over 21 results (Table 4).

6.2 Results

Results of Belief Theory Classification. Table 3 presents the classification rates for the frames of the Hammal-Caplier test database. The correct expressions and the recognized expressions are given in the first column and the first row, respectively.

Expressions E_1 (*joy*), E_2 (*surprise*) and E_4 (*neutral*) yield to good classification rates. On the contrary, the classification rate E_3 *(disgust*) is lower. This is due to individual variability (Fig. 7.a) and to the difficulty for a non actor people to simulate this expression (Fig. 7.b).

For E_1, there is a high rate of total doubt between E_1 and E_3 : the system is sure that it is one of the two expressions but is not able to know which one. This has to be related to the definition of Table 1 with two possible different states for a given

distance. In the Hammal-Caplier database, the *unknown* state E_5 often appears for intermediate frames where the person is neither in a *neutral* state, nor in a particular expression (Fig.7.c).

Table 3. Classification rates on the Hammal-Caplier database

Syst\Exp	E_1	E_2	E_3	E_4
E_1 joy	76.36	0	9.48	3
E_2 surprise	0	84.44	0	0
E_3 disgust	0	0	43.10	2
$E_1 \cup E_3$	10.90	0	8,62	0
E_4 neutral	6.66	0.78	15.51	88
E_5 unknown	6.06	11.8	12.06	0
Total	**87.26**	**84.44**	**51.2**	**88**

In order to choose between *joy* and *disgust* in case of doubt, we add a post-processing state which takes into account information about transient features and mouth shape (Sect 2). Nasal root wrinkles (Fig. 2.a) are characteristic for *disgust*. This is used to solve the problem of doubt between *joy* and *disgust*. In the case of absence of transient features, we use the ratio between length and width of the mouth (Fig. 2.b, 2.c). Our analysis shows that this ratio is larger than its value for the *neutral* expression in the case of *joy* and lower in the case of *disgust*. With the proposed post-processing step the recognition rate for E_1 (*joy*) increases by 15% and $E_1 \cup E_3$ (*joy-disgust*) decreases by 17% (2% of false detection of *disgust*). We increase by 19% for E_3 (*disgust*) and $E_1 \cup E_3$ (*joy-disgust*) decreases by 11% (5% of false detection of *joy*).

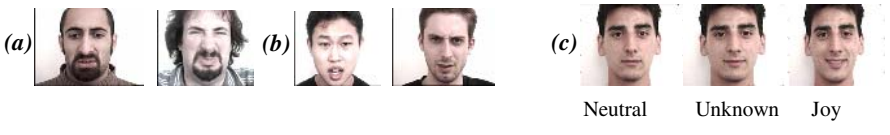


Fig. 7. Examples of *disgust* expressions : (a) individual variability; (b) poor simulations. (c): Example of *unknown* state: 3 consecutive frames from *neutral* to *joy*.

Given the fact that the state of doubt *joy-disgust* is related to the rules defined in the Table 1, it is not due to classification errors of the proposed system. It is thus possible to consider it as a good classification and to associate it to the corresponding expression which allows us to add their respecting rates leading to the results of the last row of Table 3.

Results of Bayesian Theory and HMM. Classification rates of Bayesian classifier are lower than those of belief theory classifier (Table 4 left). The best results are those of the *neutral* expression. A very low rate of classification is noted on the whole set of expressions. This is due on the one hand to the fact that the Bayesian classifier assumes specific form of the statistical distributions of the classes, which may be a wrong assumption for our dataset and on the other hand to the lack of training data.

The classification rates of the HMM are comparable with those of the belief theory (Table 4 right). Similarly the classification rates of *disgust* are better than those of *joy*, *surprise* and *neutral*.

To model the *unknown* expression used in the BeT for the HMM and the Bayesian classifiers, we introduce an "*unknown state*" which gathers all the expressions that

correspond to a set of configurations of distance states learned by the two systems as being *unknown,* contrary to the belief theory where the *unknown* expression corresponds to all the configuration of distances states unknown to the system. In other terms this is another finite set of facial expressions added to the four already defined ones and so does not contain all the possible facial configurations which can lead to classification errors. This is not the case for the belief theory which directly affects new configurations at *unknown* expression.

Table 4. Classification rates on Hammal-Caplier database. left: Bayesian classifier; right HMM classifier.

	Bayesian					HMM				
Syst\Exp	Joy	Surprise	disgust	neutral	unknown	joy	Surprise	disgust	neutral	unknown
E_1 joy	37.71	3.80	21.86	5.46	23.96	78.87	0	2.02	3.28	14.45
E_2 surprise	22.27	50.43	3.79	4.94	19.17	0	79.81	0	6.36	21.31
E_3 disgust	4.33	10.16	25.43	5.20	12.79	6.82	0	49.39	3.60	28.66
E_4neutral	7.62	20.47	2.21	79.85	24.56	4.48	10.75	6.37	75.25	25.84
E_5unknown	28.06	15.12	46.69	4.53	19.50	9.82	9.43	42.21	11.49	9.72
Total	**37.71**	**50.43**	**25.43**	**79.85**	**19.50**	**78.87**	**79.81**	**49.39**	**75.25**	**9.72**

The classification results of the three classifiers on the same data (characteristic distances) shows that the better results are obtained with the classifier based on BeT. In addition to its capacity of generalization, the use of the BeT emphasizes the fact that some expressions are not always dissociable (*joy* and *disgust*) and allows to recognize a mixture of facial expressions contrary to the HMM or Bayesian classifiers. For all these reasons we conclude that the BeT is better adapted to the problem of facial expressions recognition.

7 Conclusion

We present a method for classification of facial expressions based on the analysis of characteristic distances computed on skeletons of expression. The results of comparison with Bayesian Theory and HMM show that the best classification rates are those of the Belief Theory. To improve the results, we can increase the number and the quality of measurements, by taking into account the explicit information about the forms of contours of the skeletons of expression in addition to the characteristic distances and by taking into account the temporal evolution of measurements.

References

1. http://www.lis.inpg.fr/pages_perso/hammal/index.htm
2. Cohn, J., Zlochower, A.J., Lien, J.J., Kanade, T.: Feature-point tracking by optical flow discriminates subtle differences in facial expression. IEEE ICFGR, N°3, (1998) 396–401.
3. Pantic, M., Rothkrantz, L.J.M.: Expert system for automatic analysis of facial expressions. IVC Vol.118. (2000) 881–2000.

4. Cohen, I., Cozman, F.G., Sebe, N., Cirelo, M.C., Huang, T.S.: Semi supervised Learning of Classifiers : Theory, Algorithms and their Application to Human-Computer Interaction“, IEEE Trans. On PAMI, Vol.26, (2004) 1553–1567.
5. Eveno, N., Caplier, A., Coulon, P.Y.: Automatic and Accurate Lip Tracking. IEEE Trans. On CSVT, Vol. 14. (2004) 706–715.
6. Hammal, Z., Caplier, A. : Eye and Eyebrow Parametric Models for Automatic Segmentation. IEEE SSIAI, Lake Tahoe, Nevada (2004).
7. Smets, PH.: Data Fusion in the Transferable Belief Model. Proc. ISIF, France (2000) 21–33.
8. Malciu, M., Preteux, F.: MPEG-4 Compliant Tracking of Facial Features in Video Sequences. Proc. EUROIMAGE, ICAV3D, Greece (2001) 108–111.
9. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, Vol.77. no.2, February (1989) 257–286.
10. Moon, T.K., Stirling, W.C.: Mathematical Methods and Algorithms for Signal Processing, Prentice-Hall, (2000).

A Neural Adaptive Algorithm for Feature Selection and Classification of High Dimensionality Data

Elisabetta Binaghi[1], Ignazio Gallo[1], Mirco Boschetti[2], and P. Alessandro Brivio[2]

[1] Dipartimento di Informatica e Comunicazione, Universita' degli Studi dell'Insubria, Varese, Italy
{elisabetta.binaghi, gallo}@uninsubria.it
[2] CNR-IREA, Institute for Electromagnetic Sensing of the Environment, Via Bassini 15, 20133 Milan, Italy
{boschetti.m, brivio.pa}@irea.cnr.it

Abstract. In this paper, we propose a novel method which involves neural adaptive techniques for identifying salient features and for classifying high dimensionality data. In particular a network pruning algorithm acting on Multi-Layer Perceptron topology is the foundation of the feature selection strategy. Feature selection is implemented within the back-propagation learning process and based on a measure of saliency derived from bell functions positioned between input and hidden layers and adaptively varied in shape and position during learning. Performances were evaluated experimentally within a Remote Sensing study, aimed to classify hyperspectral data. A comparison analysis was conducted with Support Vector Machine and conventional statistical and neural techniques. As seen in the experimental context, the adaptive neural classifier showed a competitive behavior with respect to the other classifiers considered; it performed a selection of the most relevant features and showed a robust behavior operating under minimal training and noisy situations.

1 Introduction

Recent applications of Pattern Recognition and in particular of Image Analysis and Classification deal with high dimensionality data.

In this context, the use of automated classification procedures is still limited by the lack of robust methods able to cope with the intrinsic complexity of high dimensionality and the consequent Hughes phenomenon, implying that the required number of labeled training samples for supervised classification increases as a function of dimensionality [1, 2]. The problem can be addressed in two complementary ways: - identify a classification model less sensitive to the Hughes phenomenon and/or - reduce the dimensionality of data and redundancies by applying feature selection strategies. Neural networks seems to be very good candidates for simultaneous feature selection and classification [3]. In view of these considerations, we designed an experimental study to investigate the robustness of a non conventional classification model when dealing with high dimensionality data. The model integrates feature selection and classification tasks in a unified framework based on adaptive techniques [4] built on the top of conventional Multi-Layer Perceptron [5].

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 753 – 760, 2005.

The model was experimentally evaluated within a Remote Sensing study aimed to classify MIVIS hyperspectral data. Robustness was evaluated in terms of performance under different training conditions and in the presence of redundant noisy bands. The model was compared with conventional statistical classifiers, Multi-Layer Perceptron and SVM.

2 Adaptive Neural Model for Feature Selection and Classification

The use of neural networks for feature extraction and selection seems promising since the ability to solve a task with a smaller number of features is evolved during training by integrating the process of learning with feature extraction (hidden neurons aggregate input features), feature selection and classification [6].

This work presents a supervised adaptive classification model built on the top of Multi-Layer Perceptron, able to integrate in a unified framework feature selection and classification stages. The feature selection task is inserted within the training process and the evaluation of feature saliency is accomplished directly by the back-propagation learning algorithm that adaptively modifies special functions in shape and position on input layer in order to minimize training error. This mechanism directly accomplishes the feature selection task within the learning stage avoiding trial and error procedures which imply multiple training runs.

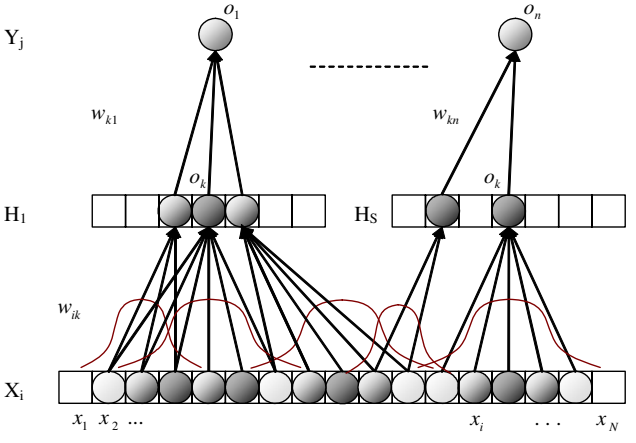


Fig. 1. Topology of the proposed adaptive Neural Model, characterized by one network for each class

Fig. 1 presents the topology of the adaptive model conceived as a composition of full connected neural networks, each of them devoted to selecting the best set of feature for discriminating one class from the others. The feature selection mechanism is embedded between input and hidden layers connections. Special functions (Figure 2a, 2b) are defined to modify connection weights: they act as penalty function for connec-

tion values and then weight the importance of features associated with the concerned input neurons.

The modified aggregation function I_k for adaptive neurons is described in the following formula

$$I_k = \sum_{i=1}^{M} w_{ik} \cdot o_i \cdot h_{ks}(i) \tag{1}$$

with M maximum number of input connections for the *j-th* neuron; o_i output of the *i-th* input neuron;

$$h_{ks}(i) = L_l(i; p, c_{ks}, a_{ks}) - L_r(i; p, c_{ks}, b_{ks}) = \frac{1}{1+e^{-p(i-(c_{ks}-a_{ks}))}} - \frac{1}{1+e^{-p(i-(c_{ks}+b_{ks}))}} \tag{2}$$

is the *s-th* bell function of the *k-th* hidden neuron; L_l and L_r are two sigmoid functions; p controls the slope of the two sigmoid functions; a_{ks} and b_{ks} controls the width of the bell function h_{ks} and c_{ks} is the centre of h_{ks}.

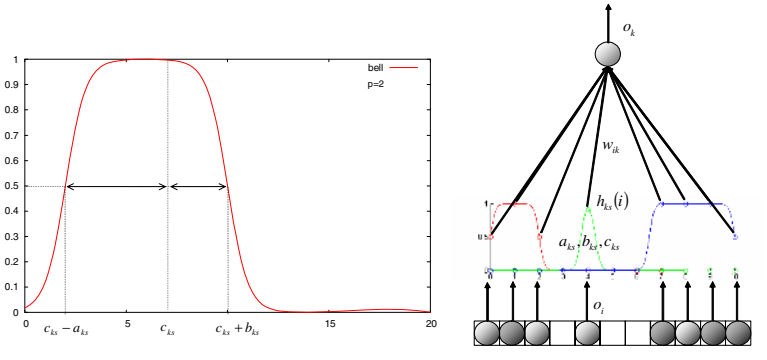


Fig. 2. (a) The bell function h_{ks} in Eq. 4 and (b) derived feature selection mechanism; $h_{ks}(i)$ represents the feature saliency measure for the *i-th* input feature

2.1 Network Configuration and Neural Learning

The neural learning procedure, aimed to identify discriminating classification functions, includes a non-conventional sub-goal formulated as the search for the most adequate number of bell functions h_{ks} varying adaptively in position and shape to lead to the smallest training error. The goal is achieved within the back-propagation learning scheme by applying the delta rule training algorithm [7] to standard weights w_{ik} and parameters a_{ks}, b_{ks} of bell functions.

At each learning step, the variation of parameters a_{ks}, b_{ks} results in a new positioning of the corresponding bell functions h_{ks}; the model attempts to re-defines the range of each bell function minimizing the overlap for all the bell functions associated with each hidden neuron; the maximum overlap allowed is at the inflection point of two adjacent bell functions.

2.1.1 Bell Function Removing and Insertion

For each bell function the distance between c_{ks} - a_{ks} and c_{ks} + b_{ks} is reduced at each learning step acting on a_{ks}, b_{ks} as follows:

$$a_{ks} = a_{ks} - r \cdot rnd \; ; \; b_{ks} = b_{ks} - r \cdot rnd \tag{3}$$

where r is a reduction rate with $0 < r \leq 1$ and rnd is a random number with $0 < rnd \leq 1$. The reduction mechanism is inserted within the overall learning process in such a way that back-propagation is able to compensate erroneous reductions.

Consequently bell functions are removed when all the following conditions are satisfied

1. $(c_{ks} + b_{ks}) - (c_{ks} - a_{ks}) < MIN\_WIDTH$, where MIN_WIDTH is a threshold value
2. $h_{ks}(m) < \dfrac{0.1}{1 + w_{mk}^2}$, where m indicates the neuron with connection weight having maximum value among those associated with connections under the bell function h_{ks}.

Variation of distance between c_{ks} - a_{ks} and c_{ks} + b_{ks} during learning can lead to a progressive increment of function areas which implies in general a decrease of connection significance. A bell function with a distance over the maximum allowed value and with mean connection weights $w_{ik} \cdot h_{ks}(i)$ under the threshold is split into two functions.

2.1.2 Removal of a Hidden Neuron

Feature selection with this type of neural net can lead to a progressive architecture simplification. In fact, as a consequence of the bell function removal mechanism, a hidden neuron can become useless for the classification task. This occurs when all the bell functions are removed by the procedure explained above, and this in turn happens when no significant connection exists between this hidden neuron and all input features.

This pruning mechanism is fundamental for training speed up and in many cases leads to a hidden layer with only the minimum number of neurons i.e. two.

An important aspect of this method is that we do not need to retrain the network after removal of a neuron and relative synapses, because the neuron was excluded by the learning procedure.

2.1.3 Initialization of the Neural Model

Initialization of the adaptive neural model involves specification of the following topological aspects:

- Number of bell functions for each neuron
- Number of neurons for each hidden layer

The proposed model is designed to cope with high dimensionality data. Considering that the number of bell functions can increase during learning by means of an insertion mechanism and that hidden neurons can be removed by the criteria stated

above, we may pose a heuristic initialization criterion which defines the minimal initial number of bell functions equal to two for each hidden neuron.

The initial number of hidden neurons can be heuristically assessed according to conventional configuration rules [5] and specifically considering, the advantage of an automatic reduction of useless hidden neurons as a function of the input dimensions.

3 Experimental Evaluation

Our experiments were designed to assess the robustness of the adaptive neural model in classifying high dimensionality data . In particular empirical tests were conducted addressing the following main questions:

- how did the performances of the neural adaptive model depends upon different levels of supervised knowledge available for training ?
- how did the neural adaptive model compare with statistical and neural classifiers ?

Experiments were conducted within a remote sensing study aimed to classify MIVIS hyperspectral data. The study area represents a typical agro-ecosystem belonging to Ticino River regional park and located south west of Milan, Italy. A detailed land cover map of this area was obtained by integrating field surveys with aerial photo interpretation thus providing labeled data for the experiments.

The source data is constituted by an hyperspectral image with a total of 102 spectral bands acquired by MIVIS (Multispectral Infrared and Visible Imaging Spectrometer) with an aerial survey over the study area. Spectral bands were reduced to 51 by eliminating noisy bands and to 92 by eliminating thermal infrared range.

Sample areas for five classes (rice, corn, bare soil, poplar, natural forest) were chosen having good spatial coverage so that the natural variability of land cover class could be ensured. Three types of sets were chosen for training, named T1, T2 and T3 having different cardinality: 100, 52 and 25 pixels respectively.

Test set was composed of 60 pixels for each cover class, randomly selected outside of training areas, by applying stratified technique, that guarantees a level of confidence, in the overall accuracy estimation, of 95% for all classes [8]

3.1 Robustness Evaluation Under Minimal Training Conditions

The experiment aimed to evaluate the performances of the proposed adaptive model when trained under three different conditions of pattern cardinality for each class.

In order to isolate factors related to training set cardinality, the overall training was facilitated by introducing a simple feature selection pre-processing stage aimed at eliminating noisy bands. The total number of features selected correspond to 51 spectral channels.

The adaptive model performed a selection of the most relevant features during training. Fig. 3 shows the bell functions assessment within the neural network topology after training with the T3 sample set, for corn and soil classes better emphasizing the feature selection mechanism. The figure exploits the situation for each of the two sub-networks devoted to the classification of a given class. The last column

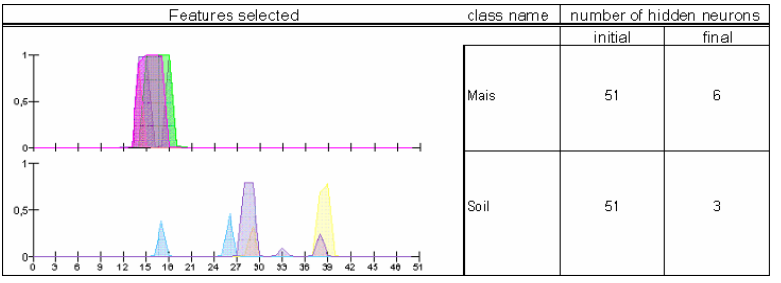


Fig. 3. Graph shows bell function assessment after training with T3 sample; columns 3 and 4 show the number of neurons for corn and soil classes, before and after training procedure

shows how the adaptive process simplifies the topology of the network: the number of initial hidden neurons was 51 for each class, but after training task many neurons were removed.

Performances of the adaptive neural model were evaluated for all the training conditions and compared with those obtained from the Maximum Likelihood (ML), the Spectral Angle Mapper (SAM) [9], a specific hyperspectral classifier, (both implemented in the software ENVI [10]), the Multi-Layer Perceptron (MLP) and the Support Vector Machine (SVM) with Radial Basis Function kernel type; parameters c and gamma were tuned using the grid tool LIBSVM [11, 12].

The agreement between reference test data and classification results was analyzed by means of the confusion matrix. Fig. 4 shows the overall accuracy (OA) values obtained for all the classifiers considered when trained with the three data sets.

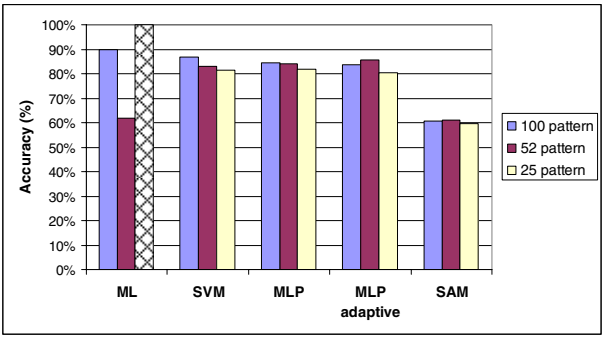


Fig. 4. Overall testing accuracy of the different methods in different pattern cardinality

The adaptive method presents a stable behavior under the three different training conditions reaching an high level of accuracy in all cases (over 80%). Maximum Likelihood is strongly influenced by the training conditions: performances are superior in case of training with T1, reaching an accuracy value close to 90% and drops to 60% using training set T2. Training T3 is not applicable. The SAM algorithm shows a stable behavior due to the fact that the classification is based on the calculation of the

distance from an average spectrum per class, but the accuracy reached is in all cases inferior to that of the adaptive model. Conventional MLP and SVM show a stable behaviour and performances comparable with those obtained by our model.

3.2 Robustness Evaluation Under a Noisy Situation

This experiment aimed to evaluate robustness of the adaptive neural model in dealing with redundant and noisy data. To this purpose the classifier was trained with the T1 data set considering all the 92 bands available (i.e. excluding thermal infrared).

Basing on positions and values of the bell functions in the trained network, we deduced the results of the feature selection procedure.

Fig. 5 shows the features selected for rice class; feature selection obtained adaptively starting from 92 bands was compared with feature selection obtained starting with pre-selected 51 bands. Results are mostly consistent. Accuracy obtained by the adaptive neural model was evaluated and compared with those obtained by the ML, MLP, and the SVM (Table 1). All the classifier registered a decrease in OA passing from 51 to 92 bands in input; however the adaptive model and ML outperformed the other three; in addition the lowest decrease was registered for the adaptive model.

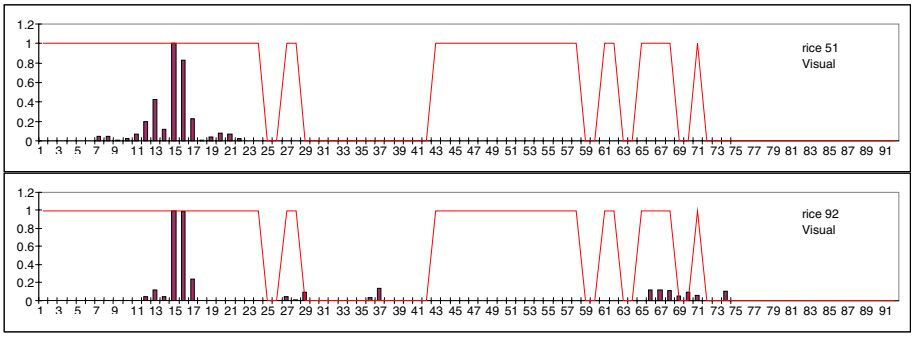


Fig. 5. Feature selection result for the rice class, compared with the preliminary band selection based on coefficient of variation

Table 1. Comparison of Adaptive MLP, ML, MLP, and SVM in terms of OA accuracy, when 51 pre-selected bands and overall 92 bands are presented in input

Number of features	51	92	
Number of training patterns	100	100	delta
ML	89.67%	76.00%	13.67%
MLP	84.67%	59.33%	25.34%
SVM	86.67%	72.00%	14.67%
AMLP	83.67%	75.33%	8.33%

4 Conclusion

In this paper, we have proposed the use of an adaptive neural network model for the twofold task of feature selection and classification. The feature selection mechanism is embedded within the back-propagation learning algorithm and directly accomplished during the learning process without implying multiple runs. Two critical aspects were investigated: minimal training and noisy situation. As seen in our experimental context, the features selection strategy allows proper selection of relevant features obtaining, as side effects, the reduction of the topological complexity of the model during training. Accuracy results obtained allow to conclude that our model can be considered an adequate tool in remote sensing studies when a feature selection phase is not possible or unadvisable and/or limited supervised data are available.

References

1. Fukunaga, K., Hayes, R.R., 1989. Effects of Sample Size Classifier Design, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no.8, pp.873-885
2. Jain A.K., Duin R.P., Mao J., 2000. Statistical Pattern Recognition: a review, IEEE Trans. on Pattern Analysis and Machine Intelligence, 22, pp.4-37
3. Jain, A., Zongker D., 1997. Feature Selection: Evaluation, Application, and Small Sample Performance, IEEE Trans. on Pattern Analysis and Machine Intelligence, 19 (2), pp.153-158.
4. Pao, Y.H., 1989. Adaptive Pattern Recognition and Neural Networks. Addison Wesley, MA
5. Bishop, C.M., 1995. Neural Networks for Pattern Recognition, Oxford University Press, Oxford.
6. Reed, R., 1993. Pruning Algorithms - a survey. IEEE Trans. Neural Networks 5, 740-747.
7. Rumelhart, H., Hinton, G.E., Williams, R.J., 1986. Learning Internal Representation by Error Propagation, Parallel Distributed Processing, Rumelhart H., Mc Lelland J.L.(eds.), 318-362. MIT Press, Cambridge, MA
8. Van Genderen J.L., Lock, B.F., Vass, P.A., 1978. Remote Sensing: Statistical testing of thematic map accuracy, Remote Sensing of Environment, 7, pp. 3-14.
9. Kruse, F. A., Lefkoff, A. B., Boardman, J. B., Heidebrecht, K. B., Shapiro, A. T., Barloon, P. J., Goetz, A. F. H., 1993. The Spectral Image Processing System (SIPS) - Interactive Visualization and Analysis of Imaging spectrometer Data, Remote Sensing of Environment, 44, pp. 145 - 163.
10. ENVI, The Environment for Visualizing Images, Research Systems Inc., http://www.rsinc.com/envi
11. Vapnik V.N., Statistical Learning Theory. Wiley, New York, 1998.
12. Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Accuracy of MLP Based Data Visualization Used in Oil Prices Forecasting Task

Aistis Raudys

Institute of Mathematics and Informatics
Akademijos 4, Vilnius 08633, Lithuania
aistis@raudys.com

Abstract. We investigate accuracy, neural network complexity and sample size problem in multilayer perceptron (MLP) based (neuro-linear) feature extraction. For feature extraction we use weighted sums calculated in hidden units of the MLP based classifier. Extracted features are utilized for data visualisation in 2D and 3D spaces and interactive formation of the pattern classes. We show analytically how complexity of feature extraction algorithm depends on the number of hidden units. Sample size – complexity relations investigated in this paper showed that reliability of the neuro-linear feature extraction could become extremely low if number of new features is too high. Visual interactive inspection of data projection may help an investigator to look differently at the forecasting problem of the financial time series.

Keywords: Feature extraction, Data mapping, Sample size, Neural network.

1 Introduction

Feature extraction, feature selection and data visualization are very important phases while solving pattern recognition (PR) and forecasting tasks. We need to reduce a number of features in order to make pattern recognition system simpler, cheaper both in a design phase and in production. Dimensionality reduction is also very important while diminishing the small sample problem: with a decrease in a number of the features the sample size / dimensionality ratio increases making the classifier design problem easier.

In numerous PR tasks, the pattern classes are not specified *a priori*. On a basis of the problem knowledge, the designer ought to specify pattern classes. An example is pattern classification of images in biomedical research where physicians often cannot specify the pattern classes, in other words, distinguish unhealthy tissues from healthy ones. Similar problems we meet in forecasting of economic time series. The Efficient Market Hypothesis [1] states that in efficient market, the prices reflect all the information available from the market. Thus, statistically significant forecast can be made only in situations where either the market is not efficient enough in terms of information processing or the forecasting method is unexpected for other participants. Possible success may be obtained in case of unexpected formulation of the forecasting problem. Original way of forming of the pattern classes could play very important

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 761 – 769, 2005.

role here. To solve this task the researcher needs to inspect statistical data and generate a number of hypothesis that support the statistical times series data at hand.

The dimensionality reduction methods can be categorized by evaluation function used to calculate the feature subset quality and a search method employed to minimize (maximize) selected performance measure. There is a vast amount of literature of feature extraction and selection, data mapping topics. It is useless to present a serious review in a short conference paper. An interested reader is referred to well known reviews, e.g. [2-4] and a recent taxonomy of data mapping methods presented in the author's paper [5].

We consider how data visualisation can be used in class formation task. We also investigate small sample size problem in the neuro-linear feature extraction, 2/3D data visualisation and neural network complexity. The hypothesis formation about possible split of the data is performed in interactive mode by visual analysis of data in 2D and 3D data mappings. A number of attempts with diverse data mapping methods should be carried out before a suitable solution is obtained.

2 Neuro-linear Feature Extraction

In this section we present formal definition of the feature extraction (FE) procedure where the complexity of FE algorithm directly depends on the number of hidden units. This FE method performs linear feature extraction with nonlinear performance criterion. This peculiarity allows to take a new look at the complexity of FE process. To find optimal data transformation weights, standard three layer MLP with additional regularization term is trained and *the first hidden layer weights* are used as coefficients in linear FE: r new features, $y_1, y_2, \dots, y_r$, are weighted sums, y_s, calculated in r hidden neurons [5]:

$$y_s = \boldsymbol{w}_s^T \boldsymbol{x} + w_{s0} \quad s = 1, 2, \dots, r, \tag{1}$$

where $\boldsymbol{w}_s^T = (w_{s0}, w_{s1}, w_{s2}, \dots, w_{sp})$ are weights of s-th *hidden* layer neuron (index "T" denotes transposition operation), p is dimensionality of the input feature space, $\boldsymbol{x}$ is p-dimensional input vector.

We see that MLP can actually be interpreted as linear mapping method with nonlinear SLP in its output stage. The new feature space depends on complexity of decision boundary. In the case of the two hidden units - two new features ($r = 2$) are obtained. The maximum complexity of decision boundary in 2D space is determined by shape "⌐¬". The complexity of DB may be higher if more than two features would be extracted. In Fig. 1a we have decision boundary (DB) of the MLP based classifier with 3 hidden neurons trained by the Levenberg-Marquardt method. It is more non-linear as "angle shaped" boundary "⌐" in case of two hidden units (boundary 1 (in dashes) in Fig. 1b). Positive side of neuro-linear feature extraction is *the possibility to control the complexity of the decision boundary*. We can: a) change the number of hidden layers, b) utilize the weight decay regularization term, c) purposefully proportionally decrease of weights, $w_{s0}, w_{s1}, w_{s2}, \dots, w_{sp}$, of each single neuron. The latter technique allows change the smoothness of decision boundary and inspect unknown structure of the data better. We will see that potential to control complexity is very important having in mind that training sample size is fixed (usually "small").

The neuro-linear feature extraction is rather similar to popular Foley – Sammon (sometimes called linear discriminant analysis feature extraction) feature mapping algorithm [6, 7] which belongs to the group of supervised methods (wrapper approach). Here the standard Fisher linear classifier (FLC) [2, 3] is used to create a linear decision boundary which helps to extract the first new feature. A new feature is a distance of vector $\boldsymbol{x}$ to discriminant hyperplane, $\boldsymbol{w}_{\mathrm{FLC}}^{T}\boldsymbol{x} + w_{\mathrm{FLC0}} = 0$, where $\boldsymbol{w}_{\mathrm{FLC}}$ and w_{FLC0} are the weights of FLC. The feature extraction procedure is repeated on the rest of the orthogonal features until there are no features left. Thus, the first extracted feature is a direction where pattern classes are best separated linearly. The second feature is the direction where classes are separated slightly worse, and so on.

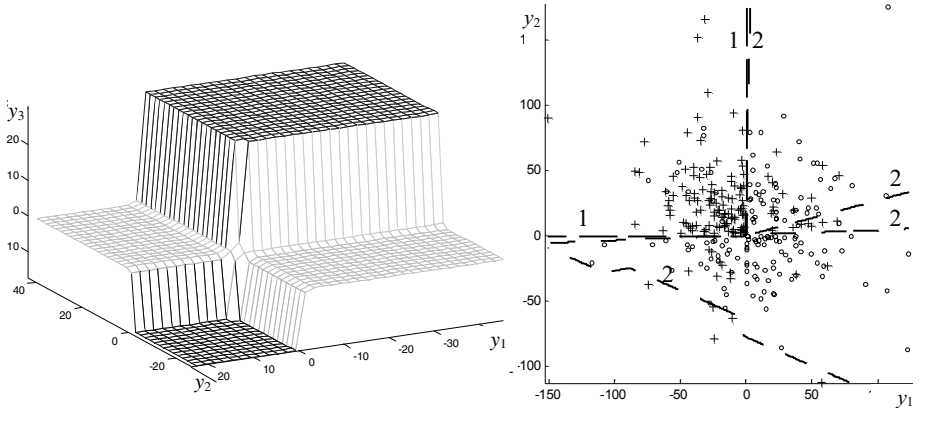


Fig. 1. Nonlinear decision boundaries of MLP: a) MLP with 3 hidden units in 3D new feature y_1, y_2 and y_3, space (resubstitution error estimate P_R= 0.12, hold out error estimate P_{HO} = 0.34). b) 1 – DB of MLP with 2 hidden units in new 2D space (P_R= 0.16,P_{HO}= 0.33), 2 – DB MLP with 5 hidden units (P_R= 0.15, P_{HO}= 0.36).

The Foley – Sammon mapping creates new features derived independently to other ones. This is disadvantage of Foley – Sammon method. Meanwhile, in neuro-linear feature extraction, *minimization of performance criterion is performed simultaneously for all new features*. In the Foley – Sammon mapping, optimization is performed sequentially: for the first new feature at first, then for the second one, etc.

Feature extraction often is utilized for data mapping into 2D or 3D spaces (data mapping) in order to perform preliminary data analysis and formation of pattern classes in interactive mode. In addition to wrapper approach, the filter approach based FE methods are used too. In most powerful of them, the projection pursuit method, one optimizes selected "indexes of interestingness" and looks for interesting low-dimensional projections of high-dimensional data.

3 Small Sample Properties

Traditionally in FE one utilizes the same data set for data mapping and for analysis of the data. If sample size is small and input dimensionality, p, is high, obviously we are

examining inappropriate projection of the data in such a case. In Fig. 1b we have seen that distribution of training vectors in new feature space shows clear data structure: the data can be separated by "Γ" shaped DB marked by 1. Disappointingly, we could not observe any structure while inspecting *test data vectors* in the same 2D space.

Regrettably, an accuracy of feature extraction was not considered in the literature yet. Feature extraction is one of the most important steps in pattern recognition, therefore, small sample properties have to be studied here too. In present section, we will apply multivariate statistical analysis technique in order to evaluate accuracy of evaluation of seperability between distinct areas of the features space belonging to different pattern classes.

In MLP training, some of the hidden layer neurons can produce outputs either very close to 0 or to 1 (if sigmoid activation function, $output_s = 1/(1+\exp(-y_s))$, is utilized in MLP training). Thus only a part of training vectors affects training process of separate hidden units. Just a subset of data influences training of each single neuron. In Fig. 1 we have seen that after training, decision boundary 1 in the new feature, $y_1, y_2, \dots, y_r$, space is composed of intersecting linear parts (lines in 2D case, planes in 3D case). Thus, for a start let us assume that in neuro-linear feature extraction, weighted sums, $y_s = \boldsymbol{w}_s^T \boldsymbol{x} + w_{s0}$, $(s = 1, 2, \dots, r)$ corresponding to each hidden unit are in fact standard Fisher classifiers formed in certain distinct portions of training data, $\Omega_1, \Omega_2, \dots, \Omega_r$. Note that under some special unrestrictive conditions in single layer perceptron (SLP) training, we obtain standard Fisher linear classifier [4, 8]. Consequently the weights of new linearly extracted features corresponding to s-th portion Ω_s of training data can be expressed as

$$\boldsymbol{w}_s = \mathbf{S}_s^{-1} (\bar{\boldsymbol{x}}_s^{(1)} - \bar{\boldsymbol{x}}_s^{(2)}), \quad w_{s0} = - \tfrac{1}{2} \boldsymbol{w}_s^T (\bar{\boldsymbol{x}}_s^{(1)} + \bar{\boldsymbol{x}}_s^{(2)}), \tag{2}$$

where $\bar{\boldsymbol{x}}_s^{(1)}$, $\bar{\boldsymbol{x}}_s^{(2)}$ and $\mathbf{S}_s$ are estimates of mean vectors and a pooled covariance matrix.

Let weighted sum (1) with weights (2) is used as new feature. Let us consider training set based estimates of *distances* $\hat{\delta}_s^1$ and $\hat{\delta}_s^2$ of the first and second class mean vectors in portion Ω_s, from discriminant boundary, $\boldsymbol{w}_s^T \boldsymbol{x} + w_{s0}, = 0$, normalized by standard deviation of $\boldsymbol{w}_s^T \boldsymbol{x} + w_{s0}$. Rather simple, however, tedious and lengthy multivariate algebra (we omit derivations in this paper) results that $\hat{\delta}_s^1$ and $\hat{\delta}_s^2$ may be expressed as

$$\hat{\delta}_s^1 = \frac{\delta_s^1 \sqrt{(1 + \frac{p}{\delta_s^2} \frac{N_{1s}+N_{2s}}{N_{1s}N_{2s}}) \frac{N_{2s}+N_{1s}}{N_{2s}+N_{1s}-p}}}{\sqrt{1 + \frac{p}{\delta_s^2} \frac{N_{1s}-N_{2s}}{N_{1s}N_{2s}}}}, \quad \delta_s^2 = \frac{\delta_s^2 \sqrt{(1 + \frac{p}{\delta_s^2} \frac{N_{1s}+N_{2s}}{N_{1s}N_{2s}}) \frac{N_{2s}+N_{1s}}{N_{2s}+N_{1s}-p}}}{\sqrt{1 + \frac{p}{\delta_s^2} \frac{N_{2s}-N_{1s}}{N_{1s}N_{2s}}}}, \tag{3}$$

where δ_s^1 and δ_s^2 are true normalized distances, N_{1s} and N_{2s} are numbers of training vectors of the first and second class in data portion Ω_s.

Derivation of Equations (3) was based on crude assumption that the Ω_s-th portion of training vectors belongs to two multivariate Gaussian distributions sharing common covariance matrix. Moreover, we ignored terms of order pN_{1s}^{-2}, pN_{2s}^{-2}.

Numerical calculation according to Eq. (3) indicates that in finite sample case, training sample based distances $\hat{\delta}_s^1$ and $\hat{\delta}_s^2$ seem larger as hypothetical true distances, δ_s^1 and δ_s^2. It means that the data could seem well separable even if in reality it is inseparable at all.

Especially dangerous is a situation when N_{1s} and N_{2s} are close to p, the input dimensionality, δ_s^1 and/or δ_s^2, being very small. Let δ_s^1=0.01 (no separation between classes), input dimensionality p=50 and N_2=N_1=145. Then $\hat{\delta}_s^1$=0.913. If only a half of training vectors affect hyperplane $\boldsymbol{w}_s^T \boldsymbol{x}+w_{s0}=0$ ($N_{2s} \approx N_{1s} \approx 72$), $\hat{\delta}_s^1$=1.45. If we have three new features and assume that only 1/4th part of the data affect hyperplane ($N_{2s} = N_{1s} = 36$), $\hat{\delta}_s^1$=2.98. Actual distance was δ_s^1=0.01. It means that from 2D mapping we conclude erroneously: "very good separation!".

Equations (3) indicate also that the bias of training sample based distances, $\hat{\delta}_s^1$ or $\hat{\delta}_s^2$, may increase in portion of training data Ω_s, where $N_{2s} >> N_{1s}$ (or $N_{2s} << N_{1s}$). Such situation is difficult to avoid in practice since there are no arguments to expect that in each subset, Ω_s, samples sizes $N_{2s} \approx N_{1s}$. Thus, in order to have a truthful picture of the data's structure, in each portion of the data, we need to have $N_{1s} >> p$ and $N_{2s} >> p$.

One more alternative is to add a noise to training vectors while extraction features by means of MLP training. Numerous simulation studies have shown that in order to minimize distortion of training data a k-nearest neighbour (colored) noise injection [9] is a useful tool. In this approach, to each training vector, say $\boldsymbol{x}_z$, one finds closest k neighbours in training set of the same pattern class. In this subspace one adds a noise and increases a number of training vectors artificially. Then one uses the new expanded data set to train MLP for subsequent feature extraction. More details can be found in [4, 9]. It is a way of introducing informal information which asserts that a space between neighbouring training vectors of one pattern class in not empty. In Fig. 2 we have an illustration obtained by means of utilization of a noise injection technique for FE. We see the same data structure both in training and test sets. For other identical conditions the test set error was reduced from P_{HO}= 0.36 without noise injection (see Fig. 1b) to 0.26. So, a noise injection approach helped us not to give up to temptation to utilize untruthful structure of the data's presented in Fig. 1b.

4 Forecasting of Huge Changes in the Oil Prices

Our studies indicated that neuro-linear linear feature extraction is very powerful tool to extract features and obtain erroneous conclusions if sample size – complexity considerations are ignored. Theory and experiment show that additional utilization of pseudo-validation set obtained by a noise injection technique can be useful in validation of the data analysis results.

In this paper the FE technique was already illustrated by commenting the classification of two pattern classes obtained from time series data describing changes in the oil prices: (a) West Texas Int. Cushing, (b) Natural Gas-Henry Hub., (c) Fuel

Oil, No.2 (NY), (d) Gasoline, Unld. Reg. Non-Oxy (NY), and (e) American Stock Exchange (AMEX) oil price index during 2923 days in a period Nov 1, 1993 - Jan 12, 2005. Forecasting of the oil prices is very important economic and almost political task which has objectives both to gain in financial bargains and assisting in creating computerized models of financial and political World development [10-12].

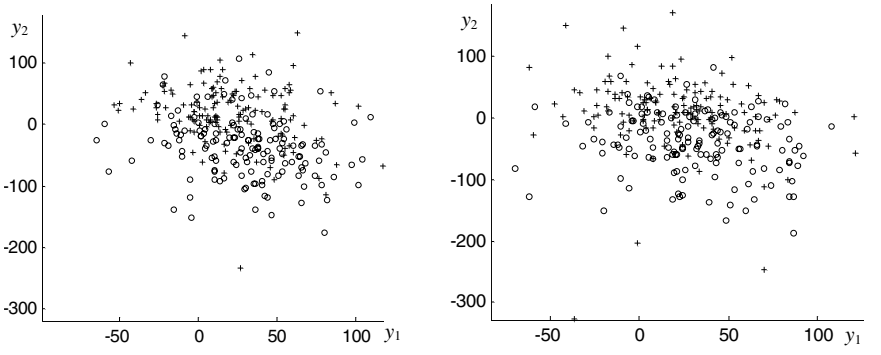


Fig. 2. Distribution of training (left) and test (right) vectors in 2D new feature, y_1, y_2, space obtained after training MLP with two hidden units with pseudo artificial training set created by noise injection procedure with: σ_n=0.5, ni_{nn}=5 and k=2 (P_R=0.23, P_{HO}=0.28 after training MLP in 50D input feature space; P_R=0.20, P_{HO}=0.26 after additional training in 2D space.

The changes in the oil prices were characterized by a difference in the AMEX. index during t_{j+1}, t_{j+2} days and subsequent t_{j+3}, t_{j+4} days. 290 days with *highest increase* in AMEX index and 290 days with *highest decrease* in this index formed two pattern class data. 50% of data were used for training and he rest for testing. Ten days history of five oil prices (the number of input features p=50) were used for classification.

The calculations according to Eq. (3) presented in previous section for 3D and high dimensional new features data have shown that apparent distances increase notably. This evaluation suggests that there is a sense to extract only two or three features. In addition we studied FE accuracy with artificial data formed from original one by means of k-NN noise injection [9]. We determined the class number of each vector by classifying this data with MLP with six hidden units.

The experiments with artificial and real data confirmed theoretical conclusion: in situations where a number of new extracted features were four or five, it was impossible to design the classifier capable discriminate the test set vectors in new feature space. Even in 2D new feature space, the best result obtained was: hold out error PHO=0.33 error while training set (resubstitution) error PR=0.16 (see Fig. 1b). In case of three new features, we succeeded to obtain 2D mappings with PR=0.06. In such cases, however, PHO was close to 50% error. In order to simplify the feature extraction procedure even more, we formed new 5 times larger pseudo artificial training set. In the regularized 2D data mapping, we did not see clear structure of the data, however, the training set and test set mappings result similar patterns (Fig. 2): the classes can be approximately separated by "Γ" shaped DB. The hold our error in

2D feature space was diminished until 0.26. Thus, proper ignorance of untruthful data structure helped us to improve classification accuracy notably. This analysis confirms earlier conclusion: often in time series prediction, simple forecasting models are more functional [13].

The weights of the neuro-linear classifier often are large and outputs of many test or training vectors are close either to zero or to one (if sigmoid activation function is used). To display the data better in new 2D space or to see differences between classification scores of pattern vectors with identical outputs sometimes it is worth to decrease the weights of each single neuron proportionally, i.e. the output of the neuron $o = f(\beta(wTx+w0))$. After some experiments, for hidden layer usually we selected $0.2<\beta<0.9$ and for output layer $0.001<\beta<0.1$. Such approach allows distinguish similar outputs and helps the end users to consider economical and political factors not included into the data. In our research, we also used principal components and projection pursuit data mapping to map artificial data mentioned above into 2D space and form two pattern classes. The supervised method, the non-linear FE, however, was more accurate for the test data discrimination.

In Fig. 3 we plotted *the test set instances* of the highest increases (circles) and decreases (stars) in the oil prices occurring in a period of 1999 – 2005. Instances of mild increase/decrease are marked by "pluses". Our results advocate that in spite of harsh competition in the oil in certain cases *it is possible to predict largest increases and decreases in the price*. The prediction accuracy may be improved by attracting informal factors into analysis, retraining of the prediction rule for every new day prediction, utilization of multiple classifier systems (see e.g. [14]).

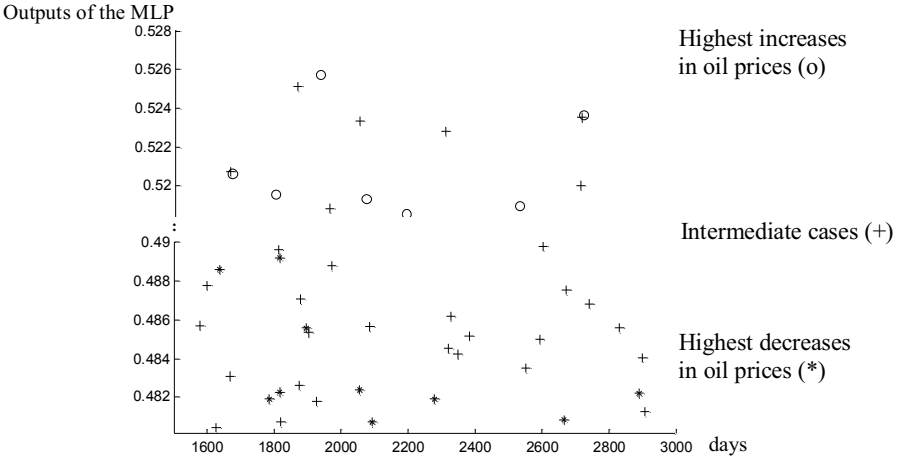


Fig. 3. Visualization of a part of classification results: specially calculated outputs of MLP classifier. Only instances of the highest increases (circles) and decreases (stars) in the oil prices occurring in a period of 1999–2005 are shown. Mild increase/decrease are marked by "pluses".

5 Conclusions

The neuro-linear feature mapping is very simple, however, useful tool for visual inspection of the data. It is a helpful researcher's adviser. To the analyst it is useful in generation of functional hypothesis about the data structure and formation of the pattern classes. The neuro-linear FE where criterion of feature quality depends on dimensionality of the new feature space. We found that an effective sample size actually participating in formation of the new features is decreasing together with an increase in dimensionality of the new feature space. Thus, in finite sample size situations, one needs to reduce a number of new features drastically. Useful tools to simplify data mapping procedure are also training the MLP classifier with artificial training set formed by means of *k*-NN noise injection and utilization of the weight decay term.

In spite of a vast research efforts undertaken in the forecasting task (see e.g. recent review [15]) and a great number of agents competing in forecasting of financial and economical time series, the prophecy may be effective if new unexpected formulation of the problem is undertaken. Informal, interactive, "visual" forecasting procedure could be of great help here. In present paper the data mapping successfully was used for formation of pattern classes in oil price forecasting task. We found that only two extracted features are reliable and useful when training history was composed of six years historical data. This recommendation was productively used also for developing adaptable forecasting algorithms of the sugar, wheat and pork belly prices [16].

Acknowledgments. The author thanks Prof. Sarunas Raudys for indication of useful literature sources and valuable suggestion concerning derivation of Equation (3).

References

[1] Fama, E.F. Efficient capital markets: A review of theory and empirical work. *Journal of Finance,* 25: 383-417, 1970.
[2] Fukunaga, K. *Introduction to Statistical Pattern Recognition*. 2nd ed. Acad. Press, NY, 1990.
[3] Duda, P.E., Hart, R.O. and Stork, D.G. *Pattern Classification*. 2nd ed. Wiley, NY, 2000.
[4] Raudys, S. *Statistical and Neural Classifiers: An integrated approach to design*. Springer, NY, 2001.
[5] Raudys, A. and J.A. Long. MLP based linear feature extraction for nonlinearly separable data. *Pattern Analysis and Applications*, 4(4): 227-234, 2001.
[6] Sammon J.W. An optimal discriminant plane. *IEEE Trans Computers,* C-19: 826-829, 1970.
[7] Foley D.H. and Sammon J.W.J. An optimal set of discriminant vectors. *IEEE Trans. on Computers*, C-24: 281-289, 1975.
[8] Koford J.S., Groner G.F. The use of an adaptive threshold element to design a linear optimal pattern classifier. *IEEE Transactions on Information Theory*, IT-2:42–50, 1966.
[9] Skurichina M., Raudys S., Duin R.P.W. K-nearest neighbors directed noise injection in multilayer perceptron training, *IEEE Trans. on Neural Networks*, 11: 504–511, 2000
[10] Adamson D.M. Soviet Gas and European-Security, *Energy Policy*, 13 (1): 13-26, 1985.

[11] Yousefi A., Wirjanto T.S. The empirical role of the exchange rate on the crude-oil price formation. *Energy Economics*, 26(5): 783-799, 2004.
[12] Leduc S., Sill K. A quantitative analysis of oil-price shocks, systematic monetary policy, and economic downturns. *J. of Monetary Economics*, 51(4): 781-808, 2004.
[13] Raudys, A. and Mockus J. Comparison of ARMA and multilayer perceptron based methods for economic time series forecasting. *Informatica*, 10(2): 231-244, 1999.
[14] Kittler J., Roli F.(eds). *Multiple Classifier Systems*. *Lecture Notes in Computer Science*, Springer-Verlag, NY, volumes 1857, 2096, 2364, 2709, 3077 (2000 – 2004).
[15] Wei H., Lai K.K., Nakamori Y., Wang S.Y. Forecasting foreign exchange rates with artificial neural networks: A review. *Int. J. on Information Technology and Decision Making*, 3(1): 145-165, 2004.
[16] Zliobaite I. and Raudys S. Prediction of commodity prices in rapidly changing environments (Lecture Notes in Computer Science ICAPR‘2005 Bath, UK, 2005).

Classification of Natural Images Using Supervised and Unsupervised Classifier Combinations

Leena Lepistö[1], Iivari Kunttu[1], Jorma Autio[2], and Ari Visa[1]

[1] Tampere University of Technology, Institute of Signal Processing,
P.O. Box 553, FI-33101 Tampere, Finland
{Leena.Lepisto, Iivari.Kunttu, Ari.Visa}@tut.fi
http://www.tut.fi/
[2] Saanio & Riekkola Consulting Engineers,
Laulukuja 4, FIN-00420 Helsinki, Finland
Jorma.Autio@sroy.fi
http://www.sroy.fi

Abstract. Combining classifiers has proved to be an effective solution to several classification problems in pattern recognition. In this paper we use classifier combination methods for the classification of natural images. In the image classification, it is often beneficial to consider each feature type separately, and combine the classification results in the final classifier. We present a classifier combination strategy that is based on classification result vector, CRV. It can be applied both in supervised and unsupervised manner. In this paper we apply our classifier combination method to the classification of rock images that are non-homogenous in terms of their color and texture properties.

1 Introduction

Considering real life classification problems it is usual that the features are spread in many difficult ways. Different classifiers may classify the same sample in different ways and hence there are differences in the decision surfaces. However, it has been found that a consensus decision of several classifiers can give better accuracy than any single classifier [1],[10].

The goal of combining classifiers is to form a consensus decision based on opinions provided by different base classifiers. The base classifiers can differ from each other in many ways [6]. In the base classifiers, there can be differences in initializations, parameter choices, architectures, classification principle, training sets, or feature sets [6]. Combined classifiers have been applied to several classification tasks, for example to the recognition of faces [14] or handwritten characters [4], person identification [3], and fingerprint verification [9].

In image classification, a number of visual descriptors are used to classify the images. In the images, there are different types of visual features, like color, texture, and shape. The feature space is typically high dimensional and the categories of images are often overlapping in the feature space. A common approach in classification is to combine all the selected descriptors into a single feature vector. The similarity between these vectors is defined using some distance metric and the most similar images

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 770 – 777, 2005.

are then classified (labeled) to the same category. However, when different types of descriptors are combined into the same feature vector, some large-scaled features may dominate the distance, while the other features do not have the same impact on the classification. Furthermore, various descriptor types use their specific distance measures in classification, which makes it even more problematic to combine these descriptors into single vector. Especially, in the case of high dimensional descriptors combination into the same vector can be problematic and may yield to remarkable drawbacks in classification performance. This is known as "the curse of dimensionality" [5]. Therefore, it is often more reasonable to consider each descriptor separately. This can be done using combined classifiers.

In this paper, we present a method to the classification of non-homogenous rock images using combined classifiers. In this method, separate base classifiers use different feature sets. The final classification can be obtained based on the combination of separate base classification results. In the base classification, supervised or unsupervised approaches can be used. In the experiments, we test these approaches in the classification of real rock images. The results are also compared to commonly used classifier combinations.

2 Combining Classifiers in Image Classification

The use of classifier combinations has been a subject of an intensive research work during last ten years. Popular solution on this field has been bagging [2], which subsamples the training data set. Another common algorithm, boosting [7], also manipulates the training data, but it emphasizes the training of samples which are difficult to classify. These methods, however, sub-sample the same feature set and therefore they cannot be applied to combination tasks with separate feature sets.

Recently, the probability-based classifier combination strategies have been popularly used in pattern recognition. Kittler et al. [10] have presented several common strategies for combining base classifiers. These strategies are e.g. product rule, sum rule, max rule, min rule, and median rule. These rules make final decision based on a posteriori probabilities generated by the base classifiers.

If the probability distributions of the base classifiers are not available, a simple combination strategy is voting. In the voting-based classifier combinations, the majority of the base classifier outputs decide the final class of the sample. In voting, the base classifiers can be regarded as black boxes, because any internal information about the base classification is not needed. Voting-based classifier combinations have been used in pattern recognition in [13].

2.1 Classification Result Vector (CRV)

The method for combining supervised classifiers is to combine the outputs of the base classifiers into a feature vector that is called classification result vector (CRV) [12]. The CRV contains the opinions of each single base classifier. Like in the voting-

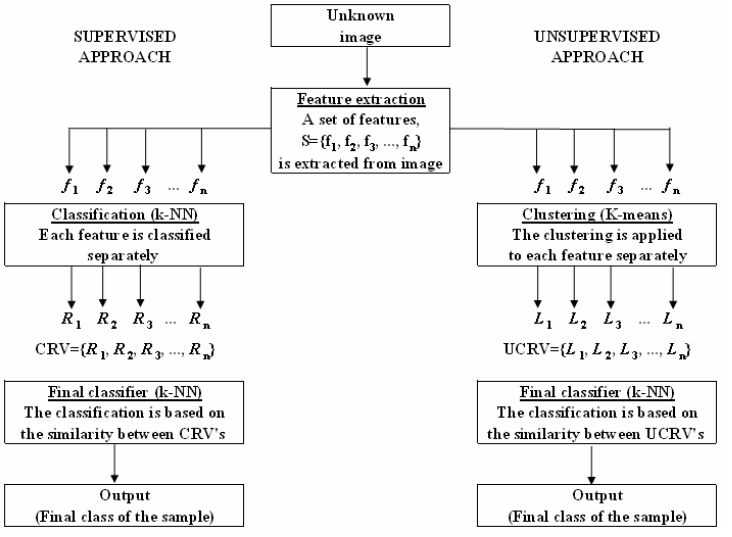


Fig. 1. The outline of the supervised and unsupervised classifier combination approaches

based approaches, also in the case of CRV method, the base classifiers can be treated as black boxes. This makes it possible to combine the classifiers without knowledge of their probability distribution. The CRV that contains the class labels provided by separate base classifiers can be used as the basis of the final classification. Hence, instead of using the majority decision of base classifier opinions, all the outputs of the base classifiers are used as the basis of final classification. This is the main difference between CRV method and voting.

The outline of the method is presented in figure 1. In this method, a set of n descriptors $S=\{f_1, f_2, f_3, \ldots f_n\}$ is extracted from an unknown sample. In the supervised approach, each feature is classified separately using supervised base classifier (e.g. k-NN classifier). The output of each base classifier is the number of class, to which the classifier assigns the sample. The class numbers are marked with R_1, R_2, R_3, ... , R_n. The outputs of the separate base classifiers are then combined by collecting them into a vector. This vector is called classification result vector (CRV), and it is used in the final classification. The final classification of the unknown sample is made based on CRV's defined for each sample. Hence, the class numbers collected into CRV's form a new feature space, in which the final classification is made. In the final classification, the similarity between the CRV's is measured using Hamming distance.

In this approach, the visual descriptors extracted from the sample image are used in classification in the base classification, which is performed for each descriptor separately. The separate base classifiers classify similar samples in similar way, which is utilized in the final classification. Hence, the CRV characterizes the behavior of the base classifiers in the case of each input pattern, which makes it possible to use CRV's to describe the content of the unknown sample. Therefore, in the final classification the samples with similar base classification results are assigned to belong to the same category. This is made by comparing the CRV's of the samples. In contrary to

voting, in CRV method the individual features do not directly affect the final classification result. Therefore, classification result is not sensitive to variations and non-homogeneities of any single features.

2.2 Unsupervised Classification Result Vector (UCRV)

Accurate classification results can also be achieved by using unsupervised base classifiers. Using unsupervised base classifiers, the base classification represents the categories, to which the unknown sample is classified without supervision. Thus the base classification obeys the natural division of each input feature. In the base classification, some clustering algorithm [8] can be used. When the clustering is performed for each feature separately, the opinions of the unsupervised base classifiers represent the natural division of each descriptor of the unknown sample. These opinions are used as features in the final classification.

Our approach to combine unsupervised base classifiers is quite similar to the CRV method presented in section 2.1. Also in this approach, n features $\{f_1, f_2, f_3, \ldots f_n\}$ are extracted from an unknown sample (figure 1). The features are then fed into a clustering algorithm (e.g. K-means clustering) separately. In addition to the input features, the algorithm is also provided with the number of the clusters (K). The clustering algorithm performs the unsupervised classification of the input features. The output of the unsupervised classifier is a set of cluster labels defined for each feature separately, $\{L_1, L_2, L_3, \ldots L_n\}$. This set represents the unsupervised classification result vector, UCRV. In the final classification, the similarity between the UCRV's is defined using Hamming distance. The final classification is also in the case of UCRV supervised.

Also in this approach, the features extracted from the sample pattern are used in classification in the separate base classifiers, which in this case are unsupervised. Because the base classification is unsupervised, any information of real classes of the samples is not used. However, the clustering algorithm assigns similar samples in the same clusters in different feature spaces. Hence the samples which are similarly clustered can be assumed to be similar. Hence, in the final classification the samples with similar UCRV's are assigned to belong to the same category.

3 Classification of Natural Images

In this part we present experiments, in which we compare our method to other, commonly used classifier combination strategies. The application field in this case is classification of natural images. The division of natural images like rock, stone, clouds, ice, or vegetation into classes based on their visual similarity is a common task in many machine vision and image analysis solutions. The classification of natural images is demanding, because they are seldom homogenous.

Rock represents typical example of non-homogenous natural image type. This is because there are often strong differences in directionality, granularity, or color of the rock texture, even if the images represented the same rock type [11]. One typical application area of the rock imaging is bedrock investigation. In this kind of analysis,

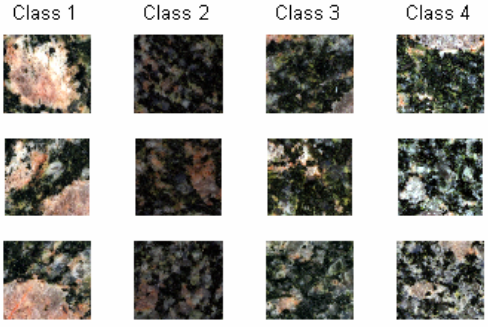


Fig. 2. Three examples from each class of the images in test set I

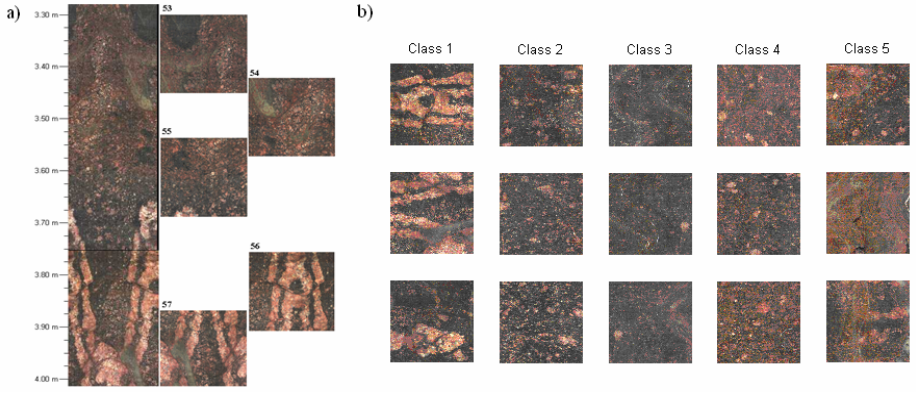


Fig. 3. a) The images of test set II are extracted from the borehole image. b) Three examples from each image class.

rock properties are analyzed by inspecting the images which are collected from the bedrock. In geological research, the rock properties are inspected based on the borehole images. Different rock layers can be recognized from the borehole images based on the color and texture properties of rock. Therefore, there is a need for an automatic classifier that is capable of classifying the rock images into visually similar classes. The use of classifier combinations has proved to be effective for this purpose [12].

As a testing database, we use two sets of rock images. Test set I consists of 336 images that are obtained by dividing large borehole images into parts. These images are manually divided into four classes by an expert. The division is based on their color and texture properties. Figure 2 presents three example images of each four class. In classes 1-4, there are 46, 76, 100, and 114 images in each class, respectively. In test set II, there are 66 non-homogenous rock images that are extracted from two boreholes. The division of a borehole image is presented in figure 3a. The example images of five types of rock in test set II are presented in figure 3b.

3.1 Classification

The classification is based on the color and texture properties of the test set images. The features used in this study consist of some MPEG-7 color and texture descriptors [16], and some other commonly used image features [12]. The color distribution of the images is characterized using color layout descriptor of MPEG-7 as well as hue and gray level histograms. In the description of texture properties of the images, homogenous texture descriptor and edge histogram descriptor of MPEG-7 are used. In addition, Gabor filtering method of Manjunath and Ma [15] is used.

In the classification experiments both supervised (CRV) and unsupervised (UCRV) approaches are tested. These approaches are compared to sum rule, max rule, median rule, and plurality voting [13], which have given good results in studies concerning classifier combinations. Product rule is not included into comparison, because the probability estimates of *k*-NN classifiers are sometimes zero, which may corrupt the result. The classification is based on the outline presented in figure 1. In the case of supervised classification (CRV), *k*-NN classification and leave one out validation methods are applied both in base classification and final classification. Unsupervised approach (UCRV) uses *K*-means clustering algorithm in the base classification, in which the value of *K* was selected to be 8, because it gave the most accurate classification. The final classification is similar to the supervised approach.

3.2 Results

The classification results are presented in tables 1a and 1b. The tables present mean classification result using separate base classifiers. In the lower part of the tables, the mean classification results using combined classifiers are presented. In this case, the classification results obtained from CVR and UCRV classification are compared to the results of the sum, max, and median rule and voting. In the classification, three values for nearest neighbors (*k*) are used. The results presented show that UCRV outperforms the other classifier combination methods in the classification of natural rock images. Both in databases I and II, the results were clearly best in the case of UCRV, whereas the CRV was the second in the comparison (database II). In the case of database I, CRV method was not able to give as accurate results, but it was still better than voting method.

4 Discussion

In this paper, we presented a method for combining classifiers in the classification of natural images. In the image classification, it is often beneficial to combine different visual descriptors to obtain the best possible classification result. Therefore, classifiers that use separate feature sets can be combined. We used natural rock images as testing material. Due to their non-homogenous nature, their classification is a difficult task.

Table 1. The classification results using separate features and classifier combinations for a) test set I and b) test set II

a)

Separate Features	Classification rate		
	k=1	k=3	k=5
ColorLayoutDescriptor	65.17 %	68.45 %	70.24 %
HomogenousTextureDescriptor	55.95 %	59.23 %	61.31 %
EdgeHistogramDescriptor	34.22 %	32.44 %	33.33 %
Gray level histogram	66.36 %	65.17 %	66.07 %
Hue histogram	69.35 %	67.56 %	65.18 %
Gabor features	61.31 %	62.80 %	61.90 %
Classifier combinations			
UCRV*	**75.00 %**	**80.65 %**	**80.36 %**
CRV	**65.18 %**	**73.51 %**	**76.78 %**
Sum rule	72.32 %	75.89 %	78.27 %
Max rule	65.18 %	74.70 %	75.00 %
Median rule	72.32 %	75.29 %	77.67 %
Voting	72.32 %	72.92 %	75.00 %

*K=8

b)

Separate Features	Classification rate		
	k=1	k=3	k=5
ColorLayoutDescriptor	59.09 %	54.55 %	57.58 %
HomogenousTextureDescriptor	46.97 %	46.97 %	46.97 %
EdgeHistogramDescriptor	31.82 %	36.36 %	36.36 %
GrayLevelHistogram	54.55 %	54.55 %	51.52 %
HueHistogram	63.64 %	62.12 %	62.12 %
Gabor features	48.48 %	53.03 %	56.06 %
Classifier combinations			
UCRV*	**80.30 %**	**72.73 %**	**71.21 %**
CRV	**69.70 %**	**63.64 %**	**68.18 %**
Sum rule	59.09 %	51.52 %	57.58 %
Max rule	59.09 %	56.06 %	56.06 %
Median rule	54.55 %	56.06 %	59.09 %
Voting	62.12 %	57.58 %	56.06 %

*K=8

The experimental results show that combined classifiers give accurate results also in the case of these kinds of images.

In our method, the feature vector that describes the image content is formed using the outputs of separate base classifiers. The class labels provided by the base classifiers form a new feature space, in which the final classification is made. In this space, the similarity between the feature vectors is defined using Hamming distance. Hence the final classification depends on the outputs of the separate classifiers, not the image features directly. This way the non-homogeneities of individual features do not have direct impact on the final result. The CRV can be formed using either supervised or unsupervised approach (UCRV). The unsupervised approach is a novel method that has proved to produce accurate classification results.

We compared the classification ability of our methods to other, commonly used classifier combination strategies in image classification. The results presented in table 1 show that our methods outperform the other classifier combination strategies in the classification of natural rock images. The results were clearly best in the case of UCRV, whereas the CRV proved also be accurate. It is also important that both the CRV and UCRV clearly outperform the voting method that is another black box approach in the classifier combination strategies.

In conclusion, the experimental results obtained from a practical image classification task show that methods presented in this paper are capable of accurate classification of real natural image databases. The methods are also computationally light and they are totally based on the outputs of separate base classifiers. Hence the base classifiers can be treated as black boxes, which make the proposed methods suitable for all kinds of classifier combinations.

Acknowledgment

The authors wish to thank Mr. Rami Rautakorpi from Helsinki University of Technology for evaluation of MPEG-7 descriptors for the test set images.

References

1. Alkoot, F.M., Kittler, J.: Experimental evaluation of expert fusion strategies, Pattern Recognition Letters, Vol. 20 (1999) 1361-1369
2. Breiman, L.: Bagging predictors. Machine Learning, Vol. 26, (1996) 123-140
3. Brunelli, R., Falavigna, D.: Person Identification Using Multiple Cues, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17 (1995) 955-966
4. Cao, J., Ahmadi, M., Shridhar, M.: Recognition of Handwritten Numerals with Multiple Feature and Multistage Classifier, Pattern Recognition, Vol. 28 (1995) 153-160
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd ed., John Wiley & Sons, New York (2001)
6. Duin, R.P.W.: The Combining Classifier: to Train or Not to Train, In: Proceedings of 16th International Conference on Pattern Recognition, Vol. 2 (2002) 765-770
7. Freund, Y., Shaphire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, Vol. 55 (1995) 119-139
8. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering, a review. ACM Computing Surveys Vol. 31 (1999) 265-323
9. Jain, A.K., Prabhakar, S., Chen, S.: Combining Multiple Matchers for a High Security Fingerprint Verification System, Pattern Recognition Letters, Vol. 20 (1999) 1371-1379
10. Kittler, J., Hatef, M., Duin, R.P.W., Matas J.: On Combining Classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20 (1998) 226-239
11. Lepistö, L., Kunttu, I., Autio, J., Visa, A.: Classification Method for Colored Natural Textures Using Gabor Filtering, In: Proceedings of 12th International Conference on Image Analysis and Processing (2003) 397-401
12. Lepistö, L., Kunttu, I., Autio, J., Visa, A.: Classification of Non-homogenous Textures by Combining Classifiers, Proceedings of IEEE International Conference on Image Processing, Vol. 1 (2003) 981-984
13. Lin, X., Yacoub, S., Burns, J., Simske, S.: Performance analysis of pattern classifier combination by plurality voting, Pattern Recognition Letters, Vol. 24 (2003) 1959-1969
14. Lu, X., Wang, Y., Jain, A.K.: Combining Classifiers for Face Recognition, In: Proceedings of International Conference on Multimedia and Expo, Vol. 3 (2003) 13-16
15. Manjunath, B.S., Ma, W.Y.: Texture Features for Browsing and Retrieval of image Data, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, (1996) 837-842
16. Manjunath, B.S., Ohm, J.-R., Vasuvedan, V.V., Yamada, A.: Color and Texture Descriptors, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11 (2001) 703-715

Estimating the ROC Curve of Linearly Combined Dichotomizers

Claudio Marrocco, Mario Molinara, and Francesco Tortorella

Dipartimento di Automazione, Elettromagnetismo,
Ingegneria dell'Informazione e Matematica Industriale,
Università degli Studi di Cassino,
03043 Cassino (FR), Italy
{c.marrocco, m.molinara, tortorella}@unicas.it

Abstract. A well established technique to improve the classification performances is to combine more classifiers. In the binary case, an effective instrument to analyze the dichotomizers under different class and cost distributions providing a description of their performances at different operating points is the Receiver Operating Characteristic (ROC) curve. To generate a ROC curve, the outputs of the dichotomizers have to be processed. An alternative way that makes this analysis more tractable with mathematical tools is to use a parametric model and, in particular, the binormal model that gives a good approximation to many empirical ROC curves. Starting from this model, we propose a method to estimate the ROC curve of the linear combination of two dichotomizers given the ROC curves of the single classifiers. A possible application of this approach has been successfully tested on real data set.

1 Introduction

Dichotomizers (i.e. two-class classifiers) are used in many critical applications (e.g., automated diagnosis, fraud detection, currency verification) which require highly discriminating classifiers. In order to improve the classification performance a well established technique is to combine more classifiers so as to take advantage of the strengths of the single classifiers and avoid their weaknesses. To this aim, a huge number of possible combination rules have been proposed up to now which generally try to decrease the classification error. However, the applications considered frequently involve cost matrices and class distributions both strongly asymmetric and dynamic and in such cases the overall error rate, usually employed as a reference performance measure in classification problems, is not a suitable metric for evaluating the quality of the classifier [1].

A more effective tool for correctly quantifying the accuracy and analyzing the dichotomizer under different class and cost distributions is given by the Receiver Operating Characteristic (ROC) curve. It provides a description of the performance of the dichotomizer at different operating points, which is independent of the prior probabilities of the two classes. ROC analysis is based in statistical decision theory and was first employed in signal detection problems [2]; it is now common in medical diagnosis and particularly in medical imaging [3]. In the Pattern Recognition field, ROC analysis is increasingly adopted for many central issues such as the evaluation

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 778 – 785, 2005.

of machine learning algorithms [4], the robust comparison of classifier performance under imprecise class distribution and misclassification costs [5] and the definition of a reject option for dichotomizers [6]. In this framework, the analysis is commonly performed on an empirical ROC curve which is generated by processing the outputs provided by the dichotomizer on a labeled data set (an efficient algorithm is described in [7]). An alternative way is to use a parametric model for the ROC curve so as to make the analysis more tractable with mathematical tools. A parametric model which gives a good approximation to many empirical ROC curves is the *binormal model*, which assumes that a pair of latent normal decision-variable distributions underlies ROC data [2, 8]. Such model plays a central role in ROC analysis, similar to the normal distribution in statistics [9].

On the basis of this model, we propose a method for estimating the ROC curve of the linear combination of two dichotomizers given the ROC curves of the single classifiers. This is a useful result since it makes possible to have an immediate preview of the performance of the system obtained by applying the combination without evaluating the outputs on the samples of the data set.

In the next section we present a short description of the binormal model and the method to obtain a parametric model of the ROC curve of the combination. The conclusive section describes a possible application of the method and shows some results obtained from experiments performed on real data sets.

2 Linear Combination of Two Classifier Based on ROC Curve

The ROC Curve correlates the *True Positive Rate* (TPR) with the *False Positive Rate* (FPR), i.e. the fraction of positive cases correctly classified with the fraction of negative cases incorrectly classified as positive. Let us define P and N as, respectively, the positive and the negative class in a two-class problem, for each fixed threshold value t, it is possible to define the rates as:

$$FPR(t) = \#\{x \in N \mid y(x) \geq t\} / \#N \qquad TPR(t) = \#\{x \in P \mid y(x) \geq t\} / \#P \tag{1}$$

where $\boldsymbol{x}$ is the considered sample and $y(\boldsymbol{x})$ is the output of the classifier. Let us, now, consider these quantities as probabilities; in this way it is possible to write:

$$\begin{aligned} FPR(t) &\cong \Pr\{y(x) \geq t \mid x \in N\} = 1 - \Pr\{y(x) < t \mid x \in N\} = 1 - F_{y|N}(t) \\ TPR(t) &\cong \Pr\{y(x) \geq t \mid x \in P\} = 1 - \Pr\{y(x) < t \mid x \in P\} = 1 - F_{y|P}(t) \end{aligned} \tag{2}$$

where $F_{y|N}(t)$ e $F_{y|P}(t)$ are two Cumulative Distribution Functions (CDF) characterizing the output of the classifier conditionally to the class of the sample x.

The binormal model is a parametric model for the ROC curve which assumes that the ROC data arise from a pair of latent normal distributions with:

$$\boldsymbol{F}_{y|N}(\boldsymbol{t}) = \Phi(\boldsymbol{t}) \quad , \quad \boldsymbol{F}_{y|P}(\boldsymbol{t}) = \Phi(\boldsymbol{bt} - \boldsymbol{a}) \tag{3}$$

where a and b are two constants and $\Phi(t)$ is the CDF of a Gaussian distribution with zero mean and unit variance. Therefore:

$$FPR(t) = 1 - \Phi(t) \quad , \quad TPR(t) = 1 - \Phi(bt - a) \tag{4}$$

A method to obtain a maximum-likelihood estimate of the parametric model from continuously distributed data has been proposed by Metz et al. in [8].

Let us now examine the classifier obtained by the linear combination of two dichotomizers: if y_1 and y_2 are the outputs of the two classifiers, the output of the combined classifier is given by $z = \alpha_1 y_1 + \alpha_2 y_2$ where α_1 and α_2 are suitably chosen weights. The TP and FP rates of this classifier are:

$$\begin{aligned} FPR_Z(t) &= \Pr\{\alpha_1 y_1 + \alpha_2 y_2 \geq t \mid x \in \mathrm{N}\} = 1 - \Pr\{\alpha_1 y_1 + \alpha_2 y_2 < t \mid x \in \mathrm{N}\} = \\ &= 1 - F_{\alpha_1 y_1 + \alpha_2 y_2 \mid \mathrm{N}}(t) \\ TPR_Z(t) &= \Pr\{\alpha_1 y_1 + \alpha_2 y_2 \geq t \mid x \in \mathrm{P}\} = 1 - \Pr\{\alpha_1 y_1 + \alpha_2 y_2 < t \mid x \in \mathrm{P}\} = \\ &= 1 - F_{\alpha_1 y_1 + \alpha_2 y_2 \mid P}(t) \end{aligned} \tag{5}$$

The two CDFs $\boldsymbol{F}_{\alpha_1 y_1 + \alpha_2 y_2 \mid \mathrm{N}}(t)$ and $\boldsymbol{F}_{\alpha_1 y_1 + \alpha_2 y_2 \mid P}(t)$ can be obtained from the CDFs of the single classifiers by means of the Stieltjes integral [10]:

$$\boldsymbol{F}_{\alpha_1 y_1 + \alpha_2 y_2 \mid \mathrm{N}}(t) = \int_{-\infty}^{+\infty} \boldsymbol{F}_{y_2 \mid \mathrm{N}}\left(\frac{t - \alpha_1 \tau}{\alpha_2}\right) d\boldsymbol{F}_{y_1 \mid \mathrm{N}}(\tau) \quad , \quad \boldsymbol{F}_{\alpha_1 y_1 + \alpha_2 y_2 \mid \mathrm{P}}(t) = \int_{-\infty}^{+\infty} \boldsymbol{F}_{y_2 \mid \mathrm{P}}\left(\frac{t - \alpha_1 \tau}{\alpha_2}\right) d\boldsymbol{F}_{y_1 \mid \mathrm{P}}(\tau) \tag{6}$$

Let us now assume that the binormal models of the ROC curves of the base classifiers are available and let their parameters be (a_1,b_1) and (a_2,b_2). This allows us to obtain a closed form of the integrals in eq. (6). To this aim, let us consider eq. (6) in terms of the corresponding density functions:

$$\begin{aligned} f_{\alpha_1 y_1 + \alpha_2 y_2 \mid \mathrm{N}}(t) &= \int_{-\infty}^{+\infty} \frac{1}{\alpha_2} f_{y_2 \mid \mathrm{N}}\left(\frac{t - \alpha_1 \tau}{\alpha_2}\right) f_{y_1 \mid \mathrm{N}}(\tau) d\tau \\ f_{\alpha_1 y_1 + \alpha_2 y_2 \mid \mathrm{P}}(t) &= \int_{-\infty}^{+\infty} \frac{1}{\alpha_2} f_{y_2 \mid \mathrm{P}}\left(\frac{t - \alpha_1 \tau}{\alpha_2}\right) f_{y_1 \mid \mathrm{P}}(\tau) d\tau \end{aligned} \tag{7}$$

Taking into account that

$$\begin{aligned} f_{y_1 \mid N}(t) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \quad f_{y_1 \mid P}(t) = \frac{b_1}{\sqrt{2\pi}} \exp\left(-\frac{(b_1 t - a_1)^2}{2}\right) \\ f_{y_2 \mid N}(t) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) \quad f_{y2 \mid P}(t) = \frac{b_2}{\sqrt{2\pi}} \exp\left(-\frac{(b_2 t - a_2)^2}{2}\right) \end{aligned} \tag{8}$$

after some algebraic manipulations, we finally obtain:

$$f_{\alpha_1 y_1 + \alpha_2 y_2 \mid N}(t) = \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{\alpha_1^2 + \alpha_2^2}} \exp\left(-\frac{t^2}{2\alpha_2^2}\left(1 - \frac{\alpha_1^2}{\alpha_1^2 + \alpha_2^2}\right)\right)$$

$$\begin{aligned} f_{\alpha_1 y_1 + \alpha_2 y_2 \mid P}(t) &= \frac{b_2 b_1}{\sqrt{2\pi}\sqrt{b_2^2 \alpha_1^2 + b_1^2 \alpha_2^2}} \exp\left(-\frac{b_2^2 t^2 - 2a_2 b_2 \alpha_2 t + \alpha_2^2\left(a_1^2 + a_2^2\right)}{2\alpha_2^2}\right) \cdot \\ &\cdot \exp\left(\frac{\left(b_2^2 \alpha_1 t - a_2 b_2 \alpha_1 \alpha_2 + a_1 b_1 \alpha_2^2\right)}{2\alpha_2^2\left(b_2^2 \alpha_1^2 + b_1^2 \alpha_2^2\right)}\right) \end{aligned} \tag{9}$$

The corresponding CDFs can now be determined:

$$F_{\alpha_1 y_1+\alpha_2 y_2|N}(t)=\frac{1}{2}\left[erf\left(\frac{t}{\sqrt{2\left(\alpha_1^{\,2}+\alpha_2^{\,2}\right)}}\right)+1\right]$$
$$F_{\alpha_1 y_1+\alpha_2 y_2|P}(t)=\frac{1}{2b}\left[erf\left(\frac{\sqrt{2}}{2}\frac{1}{\sqrt{b_2^{\,2}\alpha_1^{\,2}+b_1^{\,2}\alpha_2^{\,2}}}\left(b_1b_2t-\left(\alpha_2a_2b_1+\alpha_1a_1b_2\right)\right)\right)+1\right] \tag{10}$$

Finally, from eq. (5) we obtain

$$FPR_z(t)=1-\frac{1}{2}\left[erf\left(\frac{t}{\sqrt{2\left(\alpha_1^{\,2}+\alpha_2^{\,2}\right)}}\right)+1\right]$$
$$TPR_z(t)=1-\frac{1}{2b}\cdot\left[erf\left(\frac{\sqrt{2}}{2}\frac{1}{\sqrt{b_2^{\,2}\alpha_1^{\,2}+b_1^{\,2}\alpha_2^{\,2}}}\left(b_1b_2t-\left(\alpha_2a_2b_1+\alpha_1a_1b_2\right)\right)\right)+1\right] \tag{11}$$

3 Experimental Results

The proposed method can be applied in many situations to be faced when linearly combining two dichotomizers. For example, if the application at hand is cost-sensitive it could be useful to know the performance of the built classification system in some regions of the ROC curve [5]. Another important operation which can take advantage of our method is the evaluation of the area under the ROC curve [11] which provides an efficient way to measure the ranking quality of the classifier obtained by combining the two dichotomizers.

In this paper we have evaluated the proposed method on the latter problem. We have employed two dichotomizers: a Support Vector Machine (SVM) with linear kernel and a Multi Layer Perceptron (MLP) with five units in the hidden layer. The former has been implemented by means of SVM^{light} tool (available at http://svmlight.joachims.org) while for the latter we used the NODElib library [12]. The training of the MLP has been performed on 10000 epochs using the back propagation algorithm with a learning rate of 0.01.

The datasets used are publicly available at the UCI Machine Learning Repository [13]; all of them have two classes and a variable number of numerical input features. The features were previously rescaled so as to have zero mean and unit standard deviation. More details are given in table 1.

Table 1. Datasets used in the experiments

Data Set	# Feature	# Sample	% Positive	% Negative	Train Set	Test Set	Valid Set
Pima Indian Diabetes	8	768	34.9	65.1	384	192	192
German Credit	24	1000	30	70	500	250	250
Contraceptive Method Choice	9	1473	57.3	42.7	737	368	368

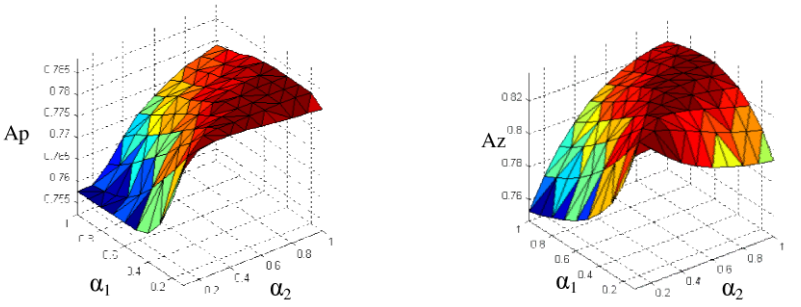
The classifier resulting from the linear combination of the two dichotomizers has output $z=\alpha_{SVM}y_{SVM}+\alpha_{MLP}y_{MLP}$, where y_{SVM} and y_{MLP} are the outputs of the two classifiers and α_{SVM} and α_{MLP} are the relative weights. We have considered a hundred of different combinations, obtained varying each of the weights from zero to one with a 0.1 step. For each dataset, once the parameters (a,b) of the binormal model of the ROC curve of each dichotomizer have been obtained by means of the method described in [8], we have generated the parametric ROC curve of the built classifier for each combination of the weights by applying eq. (11). At the same time, an empirical estimate of the same curves have been generated by means of an algorithm described in [7]. For each combination, the area under the empirical curve (Az) and the area under the parametric curve (Ap) have been compared. To avoid any bias in the comparison, 12 runs of a multiple hold-out procedure were performed on all datasets. In each run, the dataset was split in three subsets: a training set (containing 50% of the samples of each class) used in the learning phase of the dichotomizers, a validation set and a test set (each containing 25% of the samples of each class); the final size of each of these sets is given in table 1. For each run, the parameters of the binormal model were estimated on the validation set, while the empirical ROC curve was obtained from the test set. The results of the comparison for the three datasets are presented in figs 1-3 and in tables 2-4. Each cell of the tables contains a value corresponding to the mean plus or minus the standard deviation of the area under the parametric and the empirical curves relative to a particular weights combination; for the sake of readability only half of the studied combinations are shown. The figures plot the mean values of Az and Ap for all the considered combinations.

In all cases the obtained results show that the points of maximum and minimum of Az correspond to the points of maximum and minimum of Ap, thus our method is able to individuate the better and the worst couple of weights of the linear combination. In this way, it is possible to make a faster computation of all the combination without evaluating the empirical ROC and so to choice the better combination without generating the ROC curve of the combined classifiers.

However, some problems can arise when the binormal model does not fit satisfactorily the empirical ROC curve. This happens when the dichotomizer does not perform very well and the empirical ROC curve presents some concavities which the binormal model is not able to fit. Some examples are shown in fig. 4.

Table 2. Results on German Credit dataset for a linear combination of an SVM with linear kernel and an MLP

α_{MLP} \ α_{SVM}		0.2	0.4	0.6	0.8	1.0
0.2	Az	**0.7789**±0.0004	**0.7827**±0.0004	**0.7829**±0.0004	**0.7829**±0.0004	**0.7826**±0.0004
	Ap	**0.8280**±0.0011	**0.8287**±0.0005	**0.8204**±0.0003	**0.8132**±0.0003	**0.8074**±0.0002
0.4	Az	**0.7701**±0.0005	**0.7789**±0.0004	**0.7825**±0.0004	**0.7827**±0.0003	**0.7827**±0.0003
	Ap	**0.8064**±0.0018	**0.8282**±0.0011	**0.8313**±0.0006	**0.8286**±0.0005	**0.8242**±0.0004
0.6	Az	**0.7628**±0.0006	**0.7743**±0.0004	**0.7789**±0.0004	**0.7811**±0.0004	**0.7822**±0.0004
	Ap	**0.7898**±0.0020	**0.8173**±0.0016	**0.8282**±0.0011	**0.8310**±0.0007	**0.8298**±0.0006
0.8	Az	**0.7589**±0.0007	**0.7701**±0.0005	**0.7763**±0.0005	**0.7789**±0.0004	**0.7805**±0.0004
	Ap	**0.7787**±0.0020	**0.8063**±0.0018	**0.8208**±0.0014	**0.8274**±0.0011	**0.8290**±0.0008
1.0	Az	**0.7578**±0.0008	**0.7661**±0.0006	**0.7728**±0.0005	**0.7767**±0.0004	**0.7789**±0.0004
	Ap	**0.7708**±0.0019	**0.7968**±0.0020	**0.8125**±0.0017	**0.8210**±0.0013	**0.8245**±0.0010

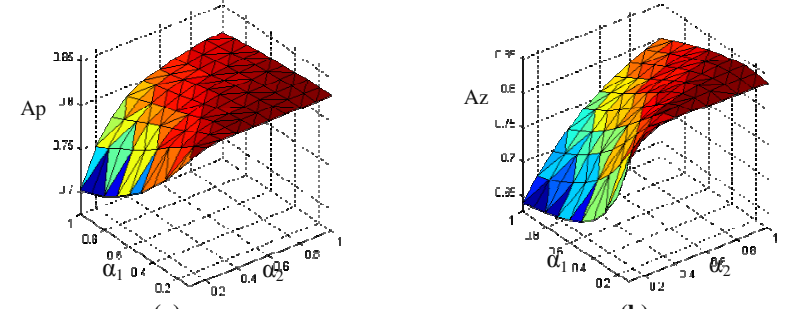


(a) **(b)**

Fig. 1. Plot of the mean values of the area under the empirical **(a)** and the parametric **(b)** ROC curve for all the considered combination on the German Credit dataset for a linear combination of an SVM with linear kernel and an MLP

Table 3. Results on Pima Indian Diabetes dataset for a linear combination of an SVM with linear kernel and an MLP

α_{MLP} \ α_{SVM}		0.2	0.4	0.6	0.8	1
0.2	Az	**0.8188**±0.0028	**0.8268**±0.0027	**0.8289**±0.0025	**0.8296**±0.0025	**0.8297**±0.0025
	Ap	**0.8016**±0.0022	**0.8348**±0.0023	**0.8406**±0.0023	**0.8414**±0.0023	**0.8408**±0.0023
0.4	Az	**0.7993**±0.0026	**0.8188**±0.0028	**0.8251**±0.0028	**0.8268**±0.0027	**0.8281**±0.0027
	Ap	**0.7410**±0.0017	**0.8016**±0.0022	**0.8252**±0.0023	**0.8347**±0.0023	**0.8385**±0.0023
0.6	Az	**0.7785**±0.0024	**0.8086**±0.0027	**0.8188**±0.0028	**0.8242**±0.0029	**0.8261**±0.0027
	Ap	**0.7057**±0.0014	**0.7677**±0.0019	**0.8016**±0.0022	**0.8195**±0.0022	**0.8288**±0.0022
0.8	Az	**0.7616**±0.0022	**0.7993**±0.0026	**0.8120**±0.0027	**0.8188**±0.0028	**0.8233**±0.0029
	Ap	**0.6845**±0.0012	**0.7410**±0.0017	**0.7781**±0.0020	**0.8012**±0.0021	**0.8150**±0.0022
1	Az	**0.7460**±0.0018	**0.7887**±0.0025	**0.8050**±0.0026	**0.8137**±0.0027	**0.8188**±0.0028
	Ap	**0.6706**±0.0011	**0.7208**±0.0015	**0.7575**±0.0018	**0.7829**±0.0020	**0.7996**±0.0021



(a) **(b)**

Fig. 2. Plot of the mean values of the area under the empirical **(a)** and the parametric **(b)** ROC curve for all the considered combination on the Pima Indian Diabetes dataset for a linear combination of an SVM with linear kernel and an MLP

Table 4. Results on Contraceptive Method Choice dataset for a linear combination of an SVM with linear kernel and an MLP

α_{MLP} \ α_{SVM}		0.2	0.4	0.6	0.8	1.0
0.2	Az	**0.7280**±0.0011	**0.7257**±0.0010	**0.7225**±0.0009	**0.7200**±0.0008	**0.7188**±0.0008
	Ap	**0.7768**±0.0014	**0.7669**±0.0011	**0.7553**±0.0010	**0.7471**±0.0010	**0.7413**±0.0009
0.4	Az	**0.7250**±0.0013	**0.7280**±0.0011	**0.7274**±0.0010	**0.7257**±0.0010	**0.7240**±0.0009
	Ap	**0.7635**±0.0015	**0.7768**±0.0014	**0.7734**±0.0012	**0.7668**±0.0011	**0.7603**±0.0011
0.6	Az	**0.7220**±0.0013	**0.7270**±0.0013	**0.7280**±0.0011	**0.7277**±0.0011	**0.7268**±0.0010
	Ap	**0.7497**±0.0015	**0.7716**±0.0015	**0.7768**±0.0014	**0.7751**±0.0013	**0.7707**±0.0012
0.8	Az	**0.7192**±0.0013	**0.7250**±0.0013	**0.7275**±0.0012	**0.7280**±0.0011	**0.7279**±0.0011
	Ap	**0.7400**±0.0015	**0.7635**±0.0015	**0.7738**±0.0014	**0.7764**±0.0013	**0.7748**±0.0013
1.0	Az	**0.7172**±0.0013	**0.7234**±0.0013	**0.7263**±0.0013	**0.7278**±0.0012	**0.7280**±0.0011
	Ap	**0.7331**±0.0014	**0.7558**±0.0015	**0.7684**±0.0015	**0.7738**±0.0014	**0.7746**±0.0013

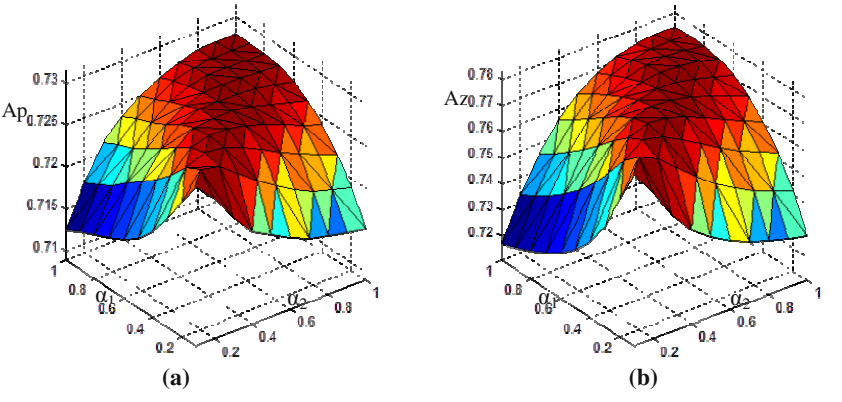


Fig. 3. Plot of the mean values of the area under the empirical **(a)** and the parametric **(b)** ROC curve for all the considered combination on the Contraceptive Method Choice dataset for a linear combination of an SVM with linear kernel and an MLP

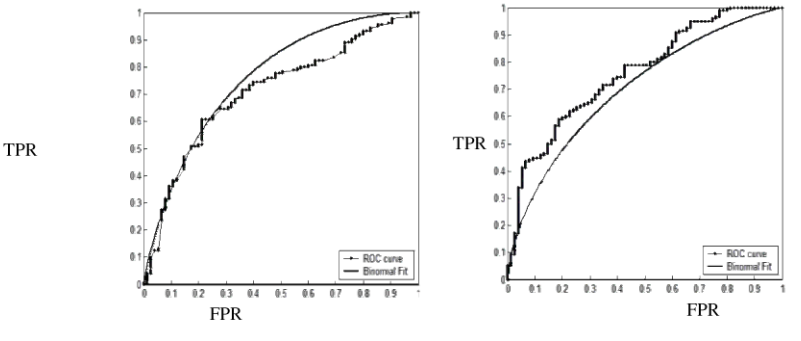


Fig. 4. Two examples of ROC curve with concavities: the fitting model does not perform well

Starting from these results, future developments of our work will examine both other algorithms for the fitting of the binormal model and other parametric models for the ROC curve.

References

[1] Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. Proc. 15th Intl. Conf. on Machine Learning, Morgan Kaufmann (1998), 445-453.
[2] Egan, J.P.: Signal detection theory and ROC analysis. Series in Cognition and Perception, Academic Press, New York (1975).
[3] Metz, C.E., ROC methodology in radiologic imaging. Invest. Radiol. 21 (1986), 720-733.
[4] Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30 (1997), 1145-1159.
[5] Provost, F., Fawcett, T., Robust classification for imprecise environments. Machine Learning 42 (2001), 203-231.
[6] Tortorella, F.: A ROC-based Reject Rule for Dichotomizers. Pattern Recognition Letters 26 (2005), 167-180.
[7] Fawcett, T.: ROC graphs: notes and practical considerations for data mining researchers. HP Labs Tech Report HPL-2003-4 (2003).
[8] Metz, C.E., Herman, B.A., Shen, J.H.: Maximum-likelihood estimation of ROC curves from continuously-distributed data. Statistics in Medicine 17 (1998), 1033-1053.
[9] Pepe, M.S. The Statistical Evaluation of Medical Tests for Classification and Prediction. Oxford Statistical Science Series, Oxford University Press (2003)
[10] Papoulis, A., Probability, Random Variables, and Stochastic Processes. McGraw-Hill, New York (2001)
[11] Cortes, C., Mohri, M.: AUC Optimization vs. Error Rate Minimization. Advances in Neural Information Processing Systems, NIPS 2003, (2004).
[12] Flake, G.W., Pearlmuter, B.A.: Differentiating Functions of the Jacobian with Respect to the Weights. In S. A. Solla, T. K. Leen, and K. Müller, eds., Advances in Neural Information Processing Systems, vol. 12, The MIT Press (2000).
[13] Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine Learning Databases. (1998) [www.ics.uci.edu/~mlearn/MLRepository.html]
[14] Metz, C.E., Pan, X.: Proper binormal ROC curves: theory and maximum-likelihood estimation. J Math Psych 43 (1999), 1-33.

Hierarchical Associative Memories: The Neural Network for Prediction in Spatial Maps

Jana Štanclová and Filip Zavoral

Department of Software Engineering,
Faculty of Mathematics and Physics, Charles University,
Malostranské nám. 25, 118 00, Praha 1, Czech Republic
{jana.stanclova, filip.zavoral}@mff.cuni.cz

Abstract. Techniques for prediction in spatial maps can be based on associative neural network models. Unfortunately, the performance of standard associative memories depends on the number of training patterns stored in the memory; moreover it is very sensitive to mutual correlations of the stored patterns. In order to overcome limitations imposed by processing of a large number of mutually correlated spatial patterns, we have designed the Hierarchical Associative Memory model which consists of arbitrary number of associative memories hierarchically grouped into several layers. In order to further improve its recall abilities, we have proposed new modification of our model. In this paper, we also present experimental results focused on recall ability of designed model and their analysis by means of mathematical statistics.

1 Introduction

The strategies for prediction in spatial maps can be based on ideas of Fukushima [2]. Let us e.g. imagine a situation when we walk through a real place known to us. In such a case, we usually see only a scenery close around us. However, we are often able to recall the scenery that we do not see yet but shall appear soon in the direction of our next movement. Triggered by the newly recalled image, we can also recall another scenery further ahead of us. Thus, we can in principle imagine the scenery of a wide area by a chain of recall processes. This ability helps us to ensure a quick and safe movement through a known environment. This process can be used e.g. in automonous driving.

Within the framework of our previous research, we studied the approach to the problem of prediction in spatial maps introduced by Fukushima [2]. The performance of associative memories is limited by number of patterns which can be stored in the model. During our cooperation with Iveta Mrázová, we have proposed the model of the so-called Hierarchical Associative Memories (HAM) which was developed with a stressed necessity to work with huge amounts of data. We expect that the HAM-model will allow a reliable and quick storage and recall of larger amounts of spatial patterns with respect to the problem of prediction in spatial maps.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 786 – 793, 2005.

2 The Associative Memories

The associative memory is a neural network, for which all its neurons are input and output neurons simultaneously and oriented interconnections are among all neurons. (Basic notions and characteristics of this memory can be found e.g. in [4]). All its weights are symmetric and each neuron is connected to all other neurons except itself. An output of the associative memory is the vector of the outputs of all the neurons in the associative memory. A weight matrix W of the associative memory with n (n>0) neurons is a n×n matrix $W=(w_{ij})$ where w_{ij} denotes the weight between the neuron i and the neuron j.

For training an associative memory, the Hebbian rule can be applied. According to this rule, the training pattern $\mathbf{x^k} = (x_1^k,..., x_n^k)$ can be stored in the associative memory with the weights w_{ij} (i,j=1,...,n) by adjusting the respective weight values w_{ij}: $w_{ij} \leftarrow w_{ij} + x_i^k x_j^k$ for i, j = 1, ... , n and $i \neq j$. We assume that the weight values w_{ij} (i, j = 1,..., n) are initialized to zero. Hence, the weight matrix $W=(w_{ij})$ corresponds to the auto-correlation matrix. For unlearning the pattern $\mathbf{x^k}$, we adjust back the respective weight values w_{ij}: $w_{ij} \leftarrow w_{ij} - x_i^k x_j^k$ for i, j = 1, ... , n and $i \neq j$. During the iterative recall, individual neurons preserve their output until they are selected for a new update. It can be shown that the associative memory with an asynchronous dynamics - each neuron is selected to update (according to the sign of its potential value ξ to +1 or -1) randomly and independently - converges to a local minimum of the energy function.

Associative memories represent a basic model applicable to image processing and pattern recognition. They can recall reliably even "damaged" patterns but their storage capacity is relatively small (approximately 0.15n where n is the dimension of the stored patterns [4]). Moreover, the stored patterns should be orthogonal or close to orthogonal one to each other. Storing correlated patterns can cause serious problems and previously stored training patterns can even become lost because the cross-talk does not average to zero [1].

3 Prediction Inspired by the Fukushima Model

In the Fukushima model [2], the chain process of predicting (recalling) the scenery of a given place far ahead is simulated using the correlation matrix memory similar to the associative memory. A "geographic map" is divided into spatial patterns overlapping each other. These fragmentary patterns are memorized in the correlation matrix memory. The actual scenery is represented in the form of a spatial pattern with an egocentric coordinate system. When we "move", the actual area "becomes shifted" relatively to our previous position in the direction of the "move" (in order to keep our body always in the center of the "scenery" pattern to be recalled). If the "scenery image" shifts following the movement of the body, a vacant region appears in the "still not seen scenery" pattern. This area is filled partially by already known pattern from previous position and partially by a vacant region from the "will not seen" part of scenery. We are trying to recall the rest of the pattern. During the recall, a pattern with a vacant "not yet seen" region (the so-called "incomplete future" pattern) is presented to the correlation matrix and the recalled pattern should fill its missing part (see Fig. 1).

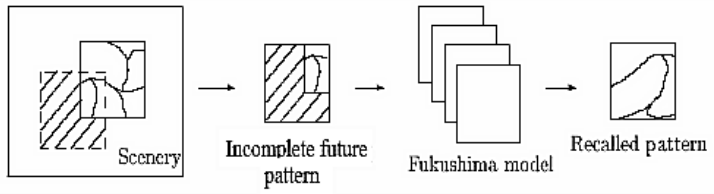


Fig. 1. Prediction inspired by the Fukushima model

Unfortunately, it is necessary to "place" the pattern presented to the correlation matrix exactly at the same location as one of the memorized patterns. The pattern to be recalled is shifted in such a way that the non-vacant region coincides with one of the memorized patterns. In order to speed-up the evaluation of the region-matching criteria, the Fukushima model incorporates the concept of the piled pattern. The point yielding the maximum correlation between the "seen scenery" and the corresponding part of the piled pattern should become the center of the next region.

The vacant part of the shifted pattern is filled, i.e. recalled by the auto-associative matrix memory. Although the recall process sometimes fails, it usually does not harm too much because the model contains the so-called monitoring circuit that detects the failure. If a failure is detected, the recalled pattern is simply discarded and recall is repeated after some time when the "body" was moved to another location.

4 The Hierarchical Associative Memory

Standard associative memories are able to recall reliably "damaged" or "incomplete" images if the number of stored patterns is relatively small and the patterns are almost orthogonal. But real patterns (and spatial maps in particular) tend to be correlated. This greatly reduces the possibility to apply standard associative memories in practice. To avoid (at least to a certain extent) these limitations, we designed (with cooperation with Iveta Mrázová) the so-called Hierarchical Associative Memory model (HAM-model). This model is based on the ideas of a Cascade Associative Memory (CASM) of Hirahara et al. [3] which allows to deal with a special type of correlated patterns. But our goal is to use the basic CASM-model more efficiently by allowing an arbitrary number of layers with more networks grouped in each layer.

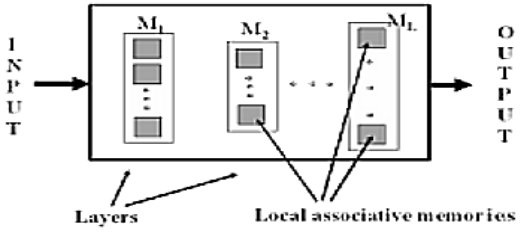


Fig. 2. Structure of the Hierarchical Associative Memory

A Hierarchical associative memory H with L layers (L>0) is an ordered L-tuple $H=(M_1,...,M_L)$ where $M_1,...,M_L$ are finite non-empty sets of associative memories; each of the memories having the same number of neurons n (n>0). A set M_k (k=1,...,L) is called layer of the memory H. $|M_k|$ denotes the number of local associative memories in the layer M_k (k=1,...,L). A training tuple T of H is an ordered L-tuple $T=(T_1,..., T_L)$ where T_k (k=1,...,L) is a finite non-empty set of training patterns for the layer M_k. The structure of the HAM-model is shown in Fig.2.

Training of the HAM-model H lies in training each of its layer M_k (k=1,...,L) separately (the so-called layer-training) and it can be done for all layers in parallel. In this way, the training patterns from the set T_k will be stored in local associative memories of the corresponding layer M_k.

During the training of the layer M_k, training patterns from the set T_k are presented to the layer M_k sequentially. For each training pattern, "the most suitable" local associative memory in the layer M_k is found and the training pattern is stored in it. If there is no "suitable" local associative memory, the new local associative memory is created and added to the layer. The pattern is stored in the newly created local memory. Now, we describe the so-called DLT-algorithm (dynamical layer training algorithm) for layer-training in formal way.

The DLT-algorithm (for the layer M_k)

1. The weight matrices of all local associative memories in M_k are set to zero.
2. A training pattern $\mathbf{x}$ from T_k is presented to the layer M_k.
3. The pattern $\mathbf{x}$ is stored in all local associative memories in the layer M_k (according to the Hebbian training rule).
4. The pattern $\mathbf{x}$ is recalled by all local associative memories from M_k. Let us denote $\mathbf{y}^i$ the output of the i-th local associative memory in the layer M_k.
5. The Hamming distance d_i of the pattern $\mathbf{x}$ and the output $\mathbf{y}^i$ is computed for each recalled output $\mathbf{y}^i$ (i=1,.. , $|M_k|$).
6. The minimum Hamming distance d_{min} is found ($d_{min} = \min \{d_i\}$, i=1,..., $|M_k|$). min is set to the index of the local associative memory with satisfying d_{min}. If there exist more local associative memories in M_k with the same minimum Hamming distance d_{min}, min will be set to the lowest index of the local memory satisfying d_{min}.
7. The pattern $\mathbf{x}$ is unlearnt from local associative memories i in the layer M_k where ($d_i \neq 0$ or $i \neq$ min).
8. If the pattern $\mathbf{x}$ is unlearnt from all local associative memories in M_k, a new local associative memory is created and added to the layer M_k. The pattern $\mathbf{x}$ is stored in the newly created local memory.
9. If there is any other training pattern in T_k, Step 2.

During recall, an input pattern $\mathbf{x}$ is presented to the HAM-model. The input pattern $\mathbf{x}$ represents an input for the first layer M_1. At every time step k ($1 \leq k \leq L$), the corresponding layer M_k produces its output ${}^k\mathbf{y}$ which is used as the input for the "next" layer M_{k+1} (i.e. ${}^{k+1}\mathbf{x} = {}^k\mathbf{y}$, $1 \leq k < L$). The output $\mathbf{y}$ of the HAM H is the output ${}^L\mathbf{y}$ of the "last" layer M_L. The recall process of the HAM-model is illustrated in Fig. 3.

Now, we focus on the recall process in one layer more precisely. During recall in the layer M_k, input pattern ${}^k\mathbf{x}$ is presented to the layer M_k. Afterwards, each local

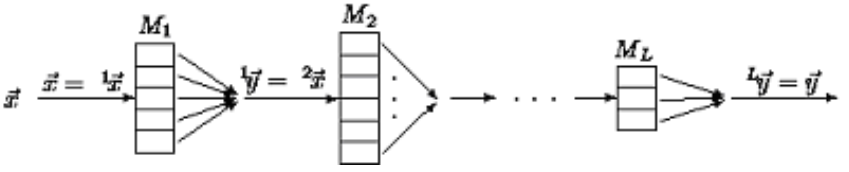


Fig. 3. Recall process in the HAM-model

associative memory in the layer M_k produces the corresponding recalled output ${}^k\mathbf{y}^i$ (i=1,..., $|M_k|$). The output ${}^k\mathbf{y}$ of the layer M_k is a recalled output which is "the most similar" to the input pattern ${}^k\mathbf{x}$. Now, we describe the so-called LR-algorithm (layer recall algorithm) for recall in one layer in formal way.

The LR-algorithm (for the layer M_k):

1. The input pattern ${}^k\mathbf{x}$ is presented to the layer M_k.
2. The pattern ${}^k\mathbf{x}$ is recalled by all local associative memories in the layer M_k. Let us denote ${}^k\mathbf{y}^i$ the output of the i-the local associative memory in the layer M_k.
3. The Hamming distance ${}^kd_i$ of the input ${}^k\mathbf{x}$ and the output ${}^k\mathbf{y}^i$ is computed for each i=1,..., $|M_k|$.
4. The minimum Hamming distance ${}^kd_{min}$ is found (${}^kd_{min}$ = min $\{{}^kd_i\}$, i=1,..., $|M_k|$). min is set to the index of the local associative memory satisfying ${}^kd_{min}$. If there exist more local associative memories with the same minimum Hamming distance ${}^kd_{min}$, min will be set to the lowest index of the local associative memory in the M_k satisfying ${}^kd_{min}$.
5. The output ${}^k\mathbf{y}$ of the layer M_k is the output ${}^k\mathbf{y}^{min}$ (i.e. ${}^k\mathbf{y} = {}^k\mathbf{y}^{min}$).

The training process starts with one local associative memory in each layer. Other local associative memories are added to the HAM-model during training according to the incoming patterns. Hence, the number of local associative memories in the HAM model depends only on the structure of training data.

Nevertheless, we should keep in mind that the above-sketched heuristic for storing patterns in the dynamically trained HAM-model is quick, simple and easy to implement but it is not optimal. Considering the DLT-algorithm, a pattern remains stored in such a local associative memory where the pattern is correctly recalled (Step 7 of the DLT-algorithm). If there is no such a local associative memory, a new local associative memory is created for storing the pattern. However, using this method for choosing the "most suitable" local associative memory, we cannot predict anything about recalling previously stored patterns. Some previously stored patterns can be recalled incorrectly (after storing some other patterns) or can even become lost.

5 Experimental Simulations and Analysis

In the previous paper [5], we presented preliminary experiments for the HAM-model and we compared the performance of the designed HAM-model with Fukushima model [2]. At this paper, we focus on robust ability of the HAM-model. In application

of the HAM-model for spatial maps prediction, the problem requires a robust recall of presented patterns, often unknown in some parts of their surface. Due to such requirements, we have proposed the following two restrictions to the HAM-model in our simulations.

1. The number of patterns which can be stored in each local associative memory of the HAM-model is limited to 0.05n, where n is the dimension of stored patterns. It corresponds approximately to 30% of the capacity for standard associative memory.

2. A pattern remains stored in that local associative memory where even its "noisy" pattern (i.e. pattern where certain number of randomly selected elements change their value to the opposite one) is recalled correctly. If there is no such local memory, a new one is created to store this pattern. Hence, it is a modification of Step 7 of the DLT-algorithm.

Anyway, the above two modifications lead in general to an increased number of local associative memories. On the other hand, the experimental results show that the second restriction does not cause rapid increase of the number of the local associative memories in the HAM-model.

The experimental simulations are restricted to a two-level hierarchy of the HAM-model. Therefore, we can call the first- and second-level patterns to be ancestors and descendants, respectively. For experiments and their further statistical analysis, we have generated 100 sets of 100 randomly generated bipolar patterns (each of size 15x15). In a bipolar pattern, every elements take the value +1 or -1. For each set of patterns, 1/4 of the patterns with the smallest cumulative correlation between the respective pattern were chosen to be the ancestors and the remaining patterns were used to form the descendants. During our experiments, we have tested "relatively small" patterns as we needed to do huge number of experiments to use statistical methods. We have performed also experiments with "bigger" data (approx. 100x100) and the results were very similar (or even a bit better).

Every experiment is run on its set of patterns independently on other sets. Experiments are repeated for every data set. During the training process, ancestors and descendants are stored in the HAM-model according to the training algorithm. During recall process, we test the HAM-model recall ability of stored patterns and their corresponding "incomplete future" patterns (of different level). A pattern is recalled correctly if it coincides with its original in "known" part. A pattern is recalled with error k if it varies with its original in k elements. For each set, we observe distribution of patterns which are correctly recalled and patterns which are recalled with error including error-rate.

First, we focus the HAM-model ability to recall stored patterns. We denote random variable Y which corresponds to ratio of correctly recalled patterns from a set of patterns. Suppose that Y has binomical distribution (number of correctly recalled patterns does not depend on other HAM-networks). For a large number of patterns, it is possible to approximate binomical distribution by a normal distribution with the same parameters.

Our experimental data leads to a null-hypothesis H: EY = 0.998 in favor of alternative hypothesis A: EY <> 0.998 at confidence level $\alpha = 0{,}05$ (EY denotes mean value of variable Y). Using statistical hypothesis testing, the hypothesis is rejected if $T \geq t_{m-1,\alpha}$ where $T = |\hat{Y} - y| / (\rho / \sqrt{n})$. According measured data, we can assess the truth of null-hypothesis H: EY=0,998 at confidence level $\alpha = 0{,}05$. Hence, we can say that the number of correctly recalled patterns is 99.8%.

Now, we focus on the HAM-model ability to recall "incomplete future" patterns. Let level k of a "incomplete future" pattern is a number of rows/columns of "incomplete" L-shaped area. During simulations, "incomplete future" of level 1, 2 and 3 are tested. Hence, a "incomplete future" pattern of the level 1 contains 13% (= 29 unknown elements / 225 total elements) "unknown" elements, the second one 25% (=56/225) "unknown" elements and the third type 36% (=81/225) "unknown" elements. The recall results are summarized in Table 1.

Table 1. The table shows a number of correctly and incorrectly recalled "incomplete future" patterns of the level 1, 2 and 3

	Level 1	Level 2	Level 3
Recalled correctly	4993	4310	3884
Recalled incorrectly	5007	5690	6116

Moreover, the distribution of incorrectly recalled "incomplete future" patterns can be analyzed by means of an error function (i.e. number of incorrectly recalled elements in one pattern). The results are shown in Fig. 4.

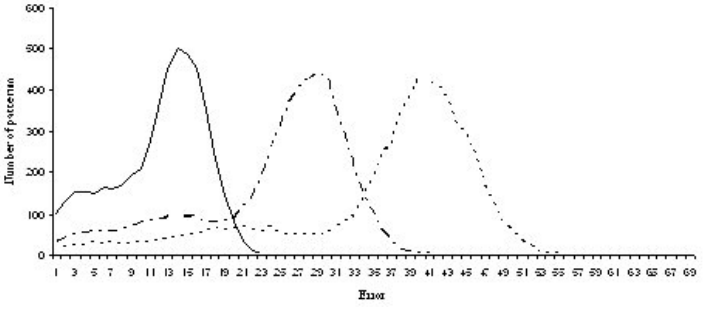


Fig. 4. Histogram of error in "incomplete future" patterns recalled incorrectly

In the figure, the axis X corresponds to a number of error in one recalled pattern and the axis Y corresponds to a number of recalled patterns with given error. The straight line depicts recall of "incomplete future" patterns of the level 1. The dash-and-dot line corresponds to recall of "incomplete future" patterns of the level 2. The dotted line denotes recall of "incomplete future" patterns of the level 3. When the "unknown" area is small, the model is able to recall such patterns quite well (the error does not exceed 10%). As the "unknown" area grows, the number of incorrectly recalled patterns is increased.

6 Conclusions

Our current research in the area of associative memories is focused on applications of associative memories for prediction in spatial maps. Unfortunately, the performance

of standard associative memories depends on the number of training patterns stored in the memory, and is very sensitive to mutual correlations of the stored patterns. In order to overcome these limitations, we have designed the Hierarchical Associative Memory model. In this paper, we presents experimental results focused on recall ability of this model.

The HAM-model improves storage ability of standard associative memories to allow to deal with large number of mutually correlated patterns. For practical application with respect to prediction problem, it is necessary to further improve robustness of the HAM-model to recall correctly (or at least with small error) patterns with larger "unknown" area. The right choice of the ancestor patterns represents an important point of a successful application of the model. We are in the process of developing more sophisticated methods - based on self-organization - for choosing "the most suitable" ancestor patterns. This could improve the robustness of the HAM-model with respect to recall patterns contained in the "unknown" area.

In the future, we plan to analyze the time- and space-complexity of the HAM-model.

References

1. D. J. Amit, H. Gutfreund, H. Sompolinsky: Information storage in neural networks with low levels of activity, in: Physical Review A, 35, 2293-2303,1987
2. K. Fukushima, Y. Yamaguchi, M. Okada: Neural Network Model of Spatial Memory: Associative Recall of Maps, in: Neural Network, Vol. 10, No. 6, pp. 971-979, 1997.
3. M. Hirahara, N. Oka, T. Kindo: A cascade associative memory model with a hierarchical memory structure, in: Neural Networks, Vol. 13, No. 1, pp. 41-50, 2000.
4. J. J. Hopfield: Neural Networks and physical system with emergent collective computational abilities, Proc. Natl. Acad. Sci. USA, 79: 2554-2558, 1982.
5. I. Mrázová, J. Tesková: Hierarchical Associative Memories for Storing Spatial Patterns, Proceedings of ITAT 2002, pp. 143-154, 2002.

Large Baseline Matching of Scale Invariant Features

Elisabetta Delponte, Francesco Isgrò, Francesca Odone, and Alessandro Verri

INFM - DISI, Università di Genova, Genova, Italy

Abstract. The problem of feature points matching between pair of views of the scene is one of the key problems in computer vision, because of the number of applications. In this paper we discuss an alternative version of an SVD matching algorithm earlier proposed in the literature. In the version proposed the original algorithm has been modified for coping with large baselines. The claim of improved performances for larger baselines is supported by experimental evidence.

1 Introduction

A number of computer vision tasks require that a set of point correspondence is established between image pairs, just to name a few: automatic feature matching is often an initialisation procedure for more complex tasks, such as fundamental matrix estimation, image mosaicing, object recognition, and three-dimensional point clouds registration.

We are currently interested in what we call *large* baseline matching, i.e. for images taken with the same camera and from not close view points. For the present work we do not aim to give a solution for the wide-baseline problem, when the images can look too much different. Our interest to this problem come from a 3D data acquisition system we are developing, for which we are dealing with the problem of registering different measures of the same object coming from different poses. Even if methods for 3D data registration do exist [1], they do need a good initialisation, that might be provided by point matched in the images, providing that an appropriate calibration between an image and a 3D point cloud is given.

We aim at obtaining a system to compute sparse correspondences between image pairs and use them to initialise a 3D registration procedure. From the feature matching standpoint our goal is to devise a procedure that allows us to obtain a reasonably high number of accurate matches from image pairs acquired by a large baseline system.

The problem of matching points between images is covered by a very rich literature. We do consider here the case when the epipolar geometry is not known, and then the corresponding point can be anywhere in the image. A classical approach, for the case of short baseline, is the one presented in [2], that adopts a standard correlation step, followed by a relaxation step. For the case of *wide* baseline we mention the work by Pritchett [3], Baumberg [4], Tuytelaars [5].

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 794–801, 2005.

Recently a simple constraints that can reduce the computational complexity for wide baseline matching, for the only case of parallel epipolar lines, has been proposed in [6].

We propose in this paper a simple modification of the SVD-based matching algorithm proposed in [7]. The core of the modification is to associate to each feature point a meaningful local descriptor to deal with larger baseline. In our method we extract from each image interesting points using difference of Gaussians, and represent them with the SIFT descriptor [8], making them tolerant to change and scale orientation. Finally our algorithm computes correspondences between the two sets of feature points with a mixed geometric and appearance based approach, as in [7]. In this last paper it is shown that a reasonably good solution to the matching problem can be achieved by an appropriate transformation of a correspondence strength matrix. The method, that builds on the top of a previous work by Scott and Longuet-Higgins [9], obtains the transformation of the strength matrix by singular value decomposition.

We report promising results, discussing how the matching behaves while the baseline grows. A comparison with the original work shows how SIFT features make the system more tolerant to the effects of a larger baseline. We also report a comparison to the matching criterion for SIFT key-points proposed by Lowe, showing that our approach is a better compromise between accurate results and a high number of correspondences.

The paper is organised as follows. In the next Section we review the SVD matching algorithm. In Section 3 the scale invariants features and their associated descriptor are briefly reviewed. Our modified SVD matching is described in Section 4, and experimental results are shown and discussed in Section 5. Section 6 is left to the final remarks.

2 SVD Matching

In this section we briefly review the matching algorithm proposed in [7]. As mentioned before this algorithm builds on the top of [9], and adapts it to deal with pixel correspondences. In the original work described in [9] it is shown that, in spite of the well-known combinatorics complexity of feature correspondence, a reasonably good solution can be achieved through the singular value decomposition of an appropriate correspondence strength matrix. In [7] this matrix is modified in order to consider intensity values together with spatial position of the feature points.

Let A and B be two images, from which we extracted m and n features respectively (A_i, $i = 1, \ldots, m$, and B_j, $j = 1, \ldots, n$), each of them represented by a simple image patch of size $w \times w$. The goal of the algorithm is to put two subsets of the two sets of features in a one-to-one correspondence.

The algorithm consists of three steps:

1. Build a correspondence matrix $\mathbf{G}$ that models both geometric proximity and similarity; each entry G_{ij} is computed as

$$G_{ij} = \frac{C_{ij} + 1}{2} e^{-r_{ij}^2 / 2\sigma^2}. \quad (1)$$

$r_{ij} = ||A_i - B_j||$ is the Euclidean distance between the two features in the image plane, and C_{ij} is the normalised correlation between the two image neighbourhood. The parameter σ controls the degree of interactions between features, where a small σ enforces local correspondences, while a bigger σ allows for more distant interactions. The elements of $\mathbf{G}$ range from 0 to 1, with higher values for more correlated features.

2. Compute the Singular Value Decomposition for $\mathbf{G}$: $\mathbf{G} = \mathbf{VDU}^\top$.
3. Compute a new correspondence matrix $\mathbf{P}$ by converting diagonal matrix $\mathbf{D}$ to a diagonal matrix $\mathbf{E}$ where each element D_{ii} is replaced with a 1: $\mathbf{P} = \mathbf{VEU}^\top$. It is shown in [9] that $\mathbf{P}$ carries similar information of $\mathbf{G}$, with the interesting property of enhancing good pairings.

In [7] experimental evidence is given that the proposed algorithm performs well on short baseline stereo pairs. In fact the performance fall as the baseline increases. It is our target to show that the reason for this behaviour is in the feature descriptor chosen and is not an intrinsic limit of the algorithm.

3 Features Detection in Scale-Space

Scale invariant features transform key-points (SIFT) were first proposed in [10] and attracted the attention of the computer vision community for their tolerance to scale changes, illumination variations, and image rotations. These features are also claimed robust to affine distortion, change of viewpoints and additive noise.

A recent comparative study presented in [11] shows that SIFT descriptors are more stable than other state of the art interest point descriptors. Therefore we decided to focus our attention on this feature descriptor, associated to the DoG key-point. In the remainder of this section we will briefly introduce the key-points and a possible description.

3.1 Scale-Space and Interest Points Detection in Images

When approaching to computer vision one of the first remarks is that every object in an image assumes a different significance if observed at a different scale. It has been demonstrated [12] that *scale-space* is a good framework to handle objects in images at different scale. Indeed, scale-space is a representation of the image which is seen at different resolution levels while its fine-scale structures are deleted.

An important aspect of the scale-space is that the suppression of details is not a random process: scale-space keeps the meaningful information obtained at the different scales.

3.2 Scale Invariant Description of Interest Points

The process of building SIFTs [8] is heavily inspired by the scale-space framework, but it keeps all the information related to the different levels of resolution.

The process can be sketched in two phases: the first is key-points detection in scale-space pyramid and the second is key-points description using the image gradient at the right level of resolution. Key-points are detected in a particular pyramidal structure that is built as follows:

1. The original image I is convolved with a Gaussian function with $\sigma = \sqrt{2}$. The result I_{σ_1} is convolved again with the same Gaussian to obtain a double-filtered image I_{σ_2}.
2. The difference between these two images is called difference-of-Gaussian (DoG): $D_{12} = I_{\sigma_1} - I_{\sigma_2}$. This will be the first level of the DoG pyramid.
3. The image is sub-sampled and the Gaussian convolution is repeated to obtain the other levels of the pyramid of DoG.

Once the pyramid of DoGs is completed, maxima and minima are located. The feature detection phase ends with a cleaning procedure for discarding low contrast features, for filtering out edges.

Regions detected by DoG are mainly blob-like structures. There are no significant signal changes in the centre of the blob and therefore the Gaussian filter-based descriptors perform better in larger point neighbourhood [11]. Once the keypoints are accurately located in the scale-space, a principal direction is assigned to each of them. This orientation is attributed in order to achieve invariance with respect to rotations.

Following [8], we associate each key-point to a descriptor, computed as a composition of direction histograms in the neighbouring regions of the scale-space, shifted according to the dominant orientation of the feature.

4 SVD Matching Using SIFT

In this section we discuss the use of the SIFT descriptor in the SVD matching algorithm. As mentioned in the introduction the SVD matching presented in [7] does not perform well when the baseline starts to increase. The reason for this behaviour is in the feature descriptor adopted. The original algorithm uses the grey level values in a neighbourhood. It is now well known that image neighbour grey level values is a descriptor too sensitive to changes in the view-point, and more robust descriptor have been introduced so far (see, for instance, [13,5,14]).

Results of a comparative study, performed on a set of planar scenes, of the performance of various features descriptors have been reported in [11], where it is shown that the SIFT descriptor is better than the other descriptors with respect rotation, scale changes, view-point change, and local affine transformations. The quality of the results decrease in the case of changes in the illumination. In the same work, cross-correlation between the image grey levels returned not stable performance, depending on the accuracy of the point, that depends strongly on the kind of transformation considered.

The considerations above suggested the use of a SIFT descriptor, instead of grey levels. We left the matrix $\mathbf{G}$ in equation (1) unchanged in its form, but C_{ij} is now the cross-correlation between SIFT descriptors. As it will be shown in

the next Section, this straightforward modification improves the performance of the SVD matching, and also gives better results, in terms of number of points correctly matched, with respect the SIFT distance used for the experiments reported in [11]. We do plan to experiment with different SIFT distances in the SVD matching, which might require to modify the form of the **G** matrix.

5 Experimental Results

In this section we report experiments carried on short image sequences where the camera was moving around a *complex* indoor scene (i.e., with several objects), increasing the baseline with respect the camera pose for the first frame in the sequence. For reason of space we show here only results on a sequence of 13 frames.

For each frame in the sequence we extracted a set of interest points, using the DoG points detector described in Section 3, that proved invariant to rotation and scale changes [10,11]. We remind here that the points detected are local scale-space extrema of the difference of Gaussians. The size of the support region, the area used for computing the associated descriptor, is determined from the selected scale.

We compare the performance of the SIFT based SVD matching, henceforth S-SVD, against the performance of the correlation based one (C-SVD), and a SIFT based point matcher proposed by Lowe in [10], and used in [11] for measuring the SIFT performance, which uses the Euclidean distance between SIFTs. We will address to this last matching method as S-DIST. More formally the correspondence are established as

- **S-SVD**: point matches are established following the algorithm of Section 2

$$C_{ij} = \sum_t \frac{(S_t^i - mean(S^i))(S_t^j - mean(S^j))}{stdv(S^i)stdv(S^j)}$$

  where S^i and S^j are the SIFT descriptors;
- **C-SVD**: point matches are determined as above but with

$$c_{ij} = \sum_t \frac{(I_t^i - mean(I^i))(I_t^j - mean(I^j))}{stdv(I^i)stdv(I^j)}$$

  where I^i and I^j are the two grey-levels neighbour;
- **S-DIST**: two features i and j matches if

$$d_{ij} = \min(D_i) < 0.6 \min(D_i - \{d_{ij}\})$$

  where $D_i = \{d_{ih} = ||S^i - S^h||\}$.

For measuring the goodness of a point match we computed the fundamental matrix [15] between the first frame and all the successive frames for each image

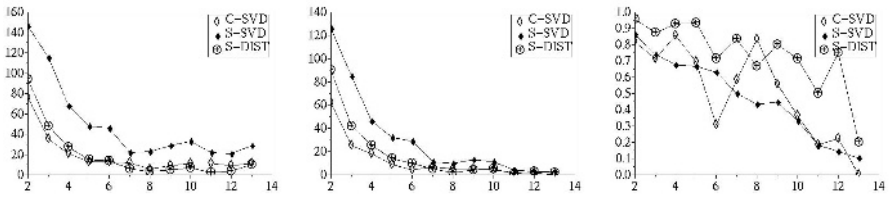


Fig. 1. Results for a 13-frames image sequence. Left: total number of matches detected. Centre: number of correct matches. Right: accuracy of the method.

Fig. 2. Results for a 13-frames image sequence. Correct matches between first (top) and sixth (bottom) frame. Left: S-SVD. Centre: C-SVD. Right: S-DIST.

sequence, using sets of point correspondences established by hand. Having the fundamental matrix $\mathbf{F}$ allows to verify if a match is correct, by using the point to epipolar line distances. We say that a match (p_i, p_j) is correct if the

$$d(p_i, \mathbf{F}p_j)^2 + d(p_j, F^t p_i)^2 < 4.$$

We would like to point out that this measure is not bullet-proof, as it can happen that wrongly matched points lie on corresponding epipolar lines. In our experience when this situation occurs, it is no more than one or two points. In Figure 3 we show stereo pair for which this happened in the case of the S-DIST matcher.

For evaluating the performance of the three point matching methods used for this work we computed: a) the total number of matches detected; b) the number of correct matches; c) the accuracy, defined as the ratio between number of correct matches and the total number of matches detected. The plot of these three values, relative to the 13-frames sequence for which we show results, can be found in Figure 1.

Overall we can say that S-SVD outperforms C-SVD in all the cases. In general S-SVD returns the largest number of correct point matches, and total number

Fig. 3. Results for a 13-frames image sequence. Correct matches between first (top) and eightieth (bottom) frame. Left: S-SVD. Centre: C-SVD. Right: S-DIST. One of the matches in the right image is wrong but the two points lie on corresponding epipolar lines, therefore is wrongly passed as correct.

of point matches. In terms of accuracy the best one is S-DIST, but S-SVD has an accuracy of more than 0.5 for almost half the length of each sequence, that makes the quality of the matches good enough for any state of the art robust estimator [16]. On the other hand, the number of correct matches detected by the S-DIST for the cases when the accuracy of the S-SVD is below 0.5, is very often too small (2-6) for being useful for any task.

The quality of the matches goes down for all the three methods as the baseline starts to be too large, meaning that none of the methods can be feasible for wide-baseline matching, and some more work needs to be done in attempting to make the S-SVD algorithm more robust to the baseline variation.

In Figure 2 and 3 we show results of the matching between the first frame of the sequence and the sixth and the eightieth frame respectively.

6 Conclusions

In this paper we described an improved version of the SVD matching presented in [7] that is capable to deal with stereo pairs with reasonably large baseline. The improvement is obtained by using a more robust feature descriptor (SIFT) than the one used in the former version of the algorithm. Experimental evidence shows the better performance of the proposed version with respect the original one, and with respect a standard SIFT based point matcher.

More work is still necessary in trying to make the algorithm feasible for wide baseline matching when the images can look too much different. In our view what should be tried is: the use of a different interest point detector as the *improved* Harris point detector discussed in [11], and the use of SIFT distance measures different form the cross-correlation used in the current version of the S-SVD.

Acknowledgements

The authors thank Maurizio Pilu for useful discussions. This work is partially supported by European FP6 NoE AIM@SHAPE grant 506766, the INFM Advanced Research Project MAIA, the FIRB Project ASTA RBAU01877P.

References

1. Besl, P., McKay, N.: A method for registration of 3D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992) 239–256
2. Deriche, R., Zhang, Z., Luong, Q.T., Faugeras, O.: Robust recovery of the epipolar geometry from an uncalibrated stereo rig. In Eklundh, J.O., ed.: Proceedings of the European Conference on Computer Vision. Volume 800 of LNCS. (1994) 567–576
3. Pritchett, P., Zisserman, A.: Wide baseline stereo matching. In: Proceedings of the International Conference on Computer Vision. (1998) 754–760
4. Baumberg, A.: Reliable feature matching across widely separated views. In: Proceedings of CVPR. (2000) 774–781
5. Tuytelaars, T., Gool, L.V.: Wide baseline stereo matching based on local, affinely invariant regions. In: Proceedings of the British Machine Vision Conference. (200) 412–425
6. Lu, X., Manduchi, R.: Wide baseline feature matching using the cross-epipolar ordering constraint. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition. Volume I. (2004) 16–23
7. Pilu, M.: A direct method for stereo correspondence based on singular value decomposition. In: Proceedings of CVPR, Puerto Rico (1997) 261–266
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
9. Scott, G., Longuet-Higgins, H.: An algorithm for associating the features of two patterns. Proc. Royal Society London **B244** (1991) 21–26
10. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the ICCV, Corfú, Greece (1999) 1150–1157
11. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: Proceedings of CVPR. Volume II. (2003) 257–263
12. Lindeberg, T.: Scale-space: A framework for handling image structures at multiple scales (1996)
13. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual corresposdence. In: Proceedings of the European Conference on Computer Vision. (1994) 151–158
14. Freeman, W., Adelson, E.: The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **13** (1991) 891–906
15. Hartley, R.I., Zisserman, A.: Multiple view geometry. Cambrige University Press (2000)
16. Meer, P., Mintz, D., Rosenfeld, A., Kim, D.Y.: Robust regression methods for computer vision: a review. International Journal of Computer Vision **6** (1991) 59–70

Landmark-Based Stereo Vision

Giovanni B. Garibotto and Marco Corvi

Elsag spa, Via Puccini, 2, 16154 Genova, Italy
giovanni.garibotto@elsag.it

Abstract. The paper refers an industrial solution to 3D measurements to be used in security application. The basic assumption of the proposed solution is the visibility of a 3D reference system in the imaged scene. The corresponding vanishing points are used as stereo point pairs at infinity, to compute the rotation between the two stereo views. To achieve a metric reconstruction, the proposed approach relies on the presence in the scene of an artificial landmark having an orthogonal triangular shape with sides of one meter length.. Reflecting cylinders are positioned at the corners and along the sides at equally spaced distances. They represent the initial stereo correspondence for system calibration. This approach has been implemented in a prototype product for the planar reconstruction in case of car accidents using a digital camera. The software application provides a simple and user-friendly interface to the process. The measure errors are always within 2 % (with a 4Mp camera).

Keywords: Computer Vision, camera calibration, 3D stereovision, perspective invariants.

1 Introduction

The paper reports an industrial solution, named LSV, Landmark-based Stereo-Vision, to solve the problem of performing distance measurements between points in 3D scenes using stereo pairs of images. This is the classical objective of photogrammetry, widely studied for building reconstruction and virtual reality rendering applications [1], [2], [3]. Our main objective is the development of effective solutions in the field of security, law-enforcement and police investigation [4]. An example of application is the metrical reconstruction of car accident scenes to speed-up the clearance of the traffic area and allow accurate measures to be performed later in a back-office application.3-D scene reconstruction from an image stereo pair is well studied from a theoretical point of view [5]. Projective stereo geometry is described by the fundamental matrix, and methods for the robust determination of this matrix have been proposed [5] [6]. The affine geometry is obtained by identifying the plane at infinity and this task is greatly simplified when a 3D reference system is detectable from the imaged scene (like building structures or road lines) [7]. Finally, in order to get a metric reconstruction, the unit lengths (in the different directions) are required. Unfortunately most natural scenes lack geometric elements for the identification of the plane at infinity and the unit lengths. Therefore, in order to achieve a metric

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 802 – 809, 2005.

reconstruction, our approach relies on the presence of an artificial landmark that provides such reference features through the use of projective invariants and the corresponding vanishing points. Our selected landmark has a right triangular shape with sides of one meter length. Reflecting cylinders are positioned at the corners and along the sides at equally spaced distances, to simplify automatic location and represent the starting points of the camera calibration process. The known lengths of the landmark sides provide a sufficient metric structure to perform 3D measures from a single view, only for 3D points lying on the landmark plane. To achieve a full metric analysis in the 3D scene it is necessary to work with a stereo pair of images. The proposed LSV system has been implemented as a software application that performs 3D measures from a pair of stereo views acquired with a standard digital camera. The application provides a simple and user-friendly interface to the process. The following section summarizes the basic approach based on the use of the triangular landmark.

2 Stereo Matching at Infinity: Description of the Approach

Binocular vision allows the metric reconstruction of the 3-dimensional geometry of a scene. This task consists of finding stereo correspondences between points in the two images, and estimating the 3D position of a point in the world from the disparity of its image points [5]. Therefore the computer vision approach to this task involves two problems, namely the identification of stereo corresponding points and stereo calibration to recover the main geometrical features. The proposed approach solves the first problem using an artificial object (landmark) located in the scene, with features that can be "easily" detected and put in correspondence. Furthermore these features are used also to perform stereo calibration. We assume a pinhole camera model, with no intrinsic distortion nor shear. The optical center (intersection of the optical axis with the image plane) is assumed to be known. The only unknown intrinsic parameter is the focal length f. A point $\boldsymbol{P}$ in 3D is projected on the image plane $\boldsymbol{p}$ by

$$\boldsymbol{P} = (X,Y,Z) \quad => \quad \boldsymbol{p} = (f\,X/Z,\ f\,Y/Z) \tag{1}$$

where (X,Y,Z) are the 3D point coordinates in the frame of reference centered on the camera, with Z along the optical axis and f is the focal length of the camera system.

In a stereo pair the two cameras have different frames of reference. The points in a stereo pair of images are related by

$$P_2 = \mathbf{R} \times P_1 + \mathbf{T} \tag{2}$$

where $\mathbf{R}$ is the rotation matrix from the first frame to the second and $\mathbf{T}$ is the translation vector (position of the first camera in the frame of the second), whose length is the baseline of the stereo configuration. This relation is simplified when the points lie at infinity,

$$P_2 = \mathbf{R} \times P_1 \tag{3}$$

$$x_2 = f\ X_2 / Z_2 = f\,(R_{xx}\,x_1 + R_{xy}\,y_1 + R_{xz}) / (R_{zx}\,x_1 + R_{zy}\,y_1 + R_{zz}) \tag{4}$$

$$y_2 = f\ \ Y_2 / Z_2 = f\,(R_{yx}\,x_1 + R_{yy}\,y_1 + R_{yz}) / (R_{zx}\,x_1 + R_{zy}\,y_1 + R_{zz})$$

ie, the image points of a point at infinity are related by a homography, namely the homography of the plane at infinity. The coefficients of this homography are the coefficients of the rotation matrix between the two camera frames. Therefore the rotation component of the stereo calibration can be determined by finding the correspondence (in the images) of points on the plane at infinity. Image points on the plane at infinity are called vanishing points because they are the images of the "center" of parallel lines in 3D. From the position of the feature elements of the landmark, vanishing points can be computed using projective invariants. Our landmark is a triangle with feature points positioned equally spaced on the sides, and at the corner. The alignment of the points, along one side, allows the estimation of the corresponding vanishing point $\boldsymbol{V_1}$, [9], using the cross ratio invariance

$$[(\boldsymbol{p_0}-\boldsymbol{p_2}) / (\boldsymbol{p_0}-\boldsymbol{p_1})]\ [(\boldsymbol{p_1}-\boldsymbol{V_1}) / (\boldsymbol{p_2}-\boldsymbol{V_1})] = 2 \tag{5}$$

since $\boldsymbol{p_1}$ is placed midway along the side of the landmark, of length **L**. The same operation is repeated for the other side of the landmark to compute the second vanishing point $\boldsymbol{V_2}$.

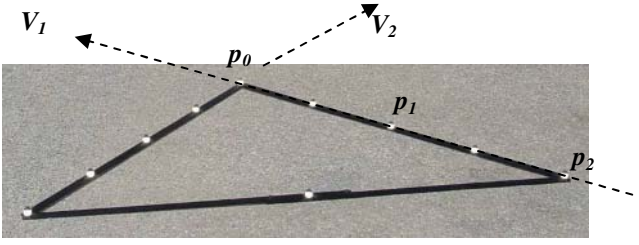


Fig. 1. The triangular landmark is placed into the scene for the automatic detection of the two orthogonal vanishing points V_1 and V_2

The directions in 3D of these two vanishing points are orthogonal, because of landmark construction, so that:

$$\boldsymbol{V_1} \bullet \boldsymbol{V_2} + f^2 = 0 \tag{6}$$

where $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$ are the image 2D vectors of the vanishing points. From this relation it is possible to compute the camera focal length. This relation can be interpreted as the equation of the line orthogonal to a vanishing point, ie, the image of the line at infinity orthogonal to a point at infinity. This is the intersection of the image plane with the plane passing through the camera center and normal to the direction of the vanishing point. One can construct the two lines orthogonal to the two vanishing points and their intersection determines a third vanishing point that corresponds to the direction perpendicular to the landmark plane. So far all the computation involves a single image, and does not require a stereo pair. It is worth noticing that this procedure gives a metric reconstruction of the plane of the landmark in 3D, ie, it allows to perform measures (distances and angles) between geometrical elements on this plane. Furthermore it is possible to compute the 3D distance D, between the camera (center O) and the landmark corner (where L is the side length of the landmark) as:

$$D = L * \sin \delta / \sin \alpha \tag{7}$$

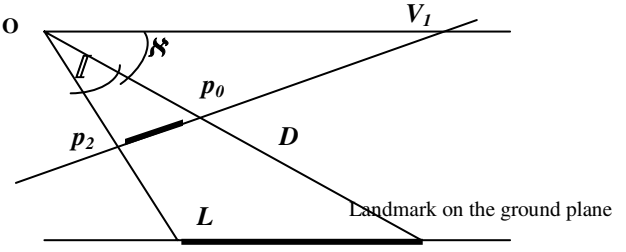


Fig. 2. Correspondence between image plane and landmark direction in the scene

The metric reconstruction with a single image extends to the third dimension (the "vertical") if the angle between the landmark sides is known. However, with a single image, there is not enough information to measure distances in the vertical direction because there is no way to determine the position of an imaged point in space, ie, its projection on the landmark plane. This information is provided by a stereo pair of images. We have two triads of orthogonal vanishing points, one in each camera frame, that correspond to the same triad of orthogonal directions in 3D. These vectors completely define the 3D rotation between the two camera frames, i.e. the homography of the plane at infinity. It is well known that the geometry of a stereo pair of images is subject to the epipolar constraint. A point in one image corresponds to a line of points in the world, and these points project on the second image in a line. The epipolar constraint says that this line must pass through a special point, the epipole, which is the projection of the first camera center in the second image. The stereo correspondence of the vanishing points and the 3D distance of the landmark corner from the two optical centers (D_1, D_2) allows to compute the baseline, i.e., the distance between the two camera centers in the world, as shown in Fig 3.

$$B^2 = D_1^2 + D_2^2 - 2\, D_1\, D_2 \cos(\beta_{12}) \tag{8}$$

β_{12} is the angle between P_2 and $P_{2\infty}$, i.e. the result of rotation $\mathbf{R} \times P_1$ in eq (3). Furthermore, the stereo pair allows a more robust calibration by minimizing the error of the stereo geometry between the two images. Keeping fixed the lines of the landmark sides (in the two images), the geometry is determined by the two vertical vanishing points and by the focal length. In fact the vertical vanishing point defines the "horizon" (line at infinity) which intersects the landmark sides in the other two vanishing points. Since the landmark sides are corresponding lines in the two images, these vanishing points are also in correspondence.
The error function to minimize consists of the following geometrical (metric and projective) constraints:

- the aspect-ratio error: i.e. the ratio of the distances between the camera center and the landmark corner computed using the two landmark side (with the single image metric);
- the orthogonality error: how far the two vanishing points on the horizon are from orthogonality (the abs cosine of their angle);

- the distance error: how much the measures of the two landmark sides differs from 1 meter (in the stereo reconstruction);
- the epipolar error: how much any three epipolar lines fail to pass through a single point (the epipole).

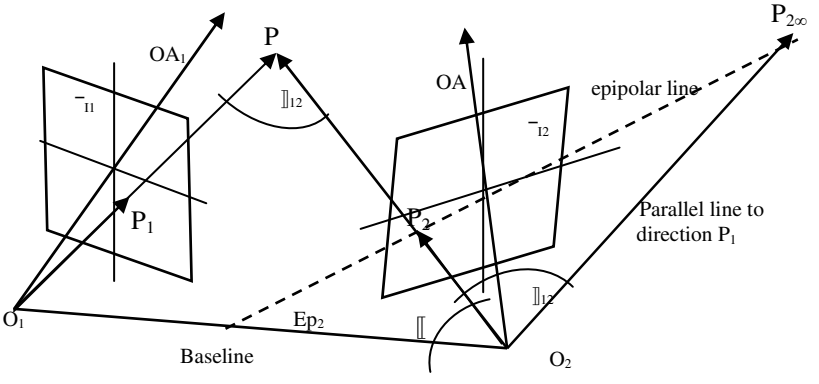


Fig. 3. Binocular stereo configuration

3 Implementation

The reconstruction approach described in the previous section has been implemented in an industrial product, for the reconstruction of the ground map of the scene in car accidents. Digital pictures of the scene are already taken by the police personnel for documentation purposes. Therefore our proposed solution, with the minor overhead of placing the reference landmark in the scene, saves the time needed for carrying on and annotating all measures on the spot, deferring this operation to a later time, in the police office. The landmark has been specially designed for easy transport (foldable) and is provided with cylinders covered with a highly reflective film, in order to ensure their visibility in every environment condition when shot by the camera flash. There are three cylinders on the short sides, besides the two at the corners, and one at the midpoint of the hypotenuse, for a total of ten cylinders. The ten cylinders form the basic element and are used to perform the stereo calibration. The sides of the landmark are low-reflecting and black, to improve the automatic detection of the cylinders. The operating procedure starts with outdoor shooting of the scene, taking a series of snaps from different points of view. A standard digital camera is used (4Mpixel) but higher resolution cameras can also be adopted, to improve reconstruction precision.

The operative constraints for the image acquisition are rather simple:

- the landmark should be positioned in the middle of the scene, not too far from the camera;
- to achieve better results, the zoom of the camera should not be changed between the shots using the same focal length, an additional constraint is applied).

The second step of the operating procedure takes place at office, where the user can run on his own PC a software tool that implements the metric reconstruction of the scene. The application software has been developed under Windows 2000/XP operating systems, with Microsoft Visual C++.

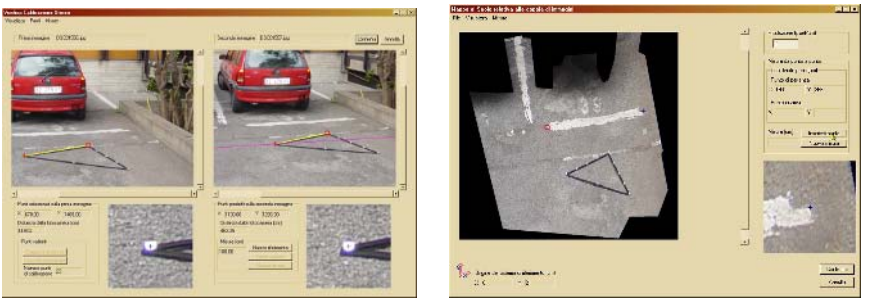


Fig. 4. a) The human interface of LSV (Landmark Stereo Vision); b) ground map

The developed human interface (an example is shown in fig.4) allows the user

- to navigate in the file-system to select image pairs, with optional thumbnail preview;
- to run landmark localization on the images through an automatic procedure (with possible fallback on a manual procedure in case of failure);
- to define the ground region of interest for each image;
- to identify points in stereo correspondence in the image pair and store them in the system;
- to show and print the map of the ground either for each single image or for both of them together;
- to add on the ground map graphic features and text of interest, like point labels;
- to perform measures between points, both working on the pair of images, and on the ground map;
- to perform reporting tasks of the measures taken during a work session;
- to save and store all the information in a database, for retrieval and use in a later session of work.

Localization of the landmark

As one might expect it turns out that a precise localization of the landmark cylinders on the images is crucial for an accurate stereo calibration. In fact this is the only necessary information to start stereo measurement. We use reflecting cylinders and all stereo pictures must be acquired with the camera flashlight on. In this way in any environmental conditions (day and night) the landmark features will appear as bright spots in the images and their detection is greatly simplified. The automatic localization of the cylinders is performed in two steps. The first step roughly localizes the positions using the high luminance level of the cylinders, by thresholding the image intensity and searching for ten blobs of suitable size (nr of pixels) and shape (roughly round). The threshold is selected automatically within a selected search area. The second step provides a sub-pixel localization through a correlation with a template model at different scales. Geometric consistency is checked to validate the

correct detection of such features. There are nevertheless situations where the automatic localization cannot succeed, e.g., when the landmark is placed on a reflecting surface (e.g. a wet road). This ambiguous situation is automatically detected by the process and the user is prompted to manually select and define the locations of the cylinders.

Calibration and ground map reconstruction

Stereo calibration can be computed using only the landmark points localized on the two images. Nevertheless, it is possible to improve and refine calibration with additional stereo pairs. Such additional points can be easily inserted and edited by the user on the application console. After calibration, the user can perform measures between points selected on the images. The ground region is a polygonal area and it is used in the construction of the ground map. The user may define the ground area by selecting the corners of any polygonal shape on the image, to remove the objects not lying on the floor. The ground map provides a top view of the scene, by merging the information from both image views. It is possible to perform measures on the ground map between manually selected points. It is also possible to project onto this map all measured 3D points to achieve a full representation of the imaged scene.

4 Results of Performance

The procedure has been tested with a set of image stereo pairs by comparing the distances measured by the LSV application software against those measured on the scene with a plastic metric tape. The reference distances are therefore affected by an error of 1% (typical error of measures with a plastic metric tape).

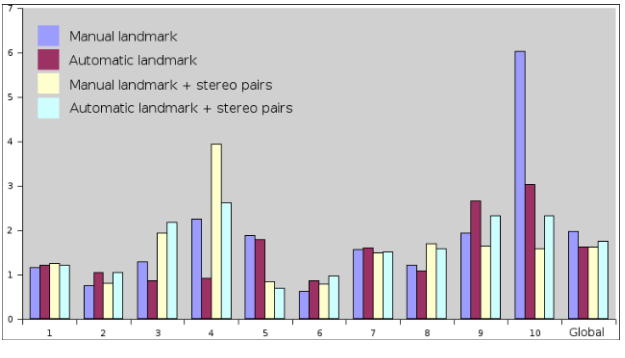


Fig. 5. 3D measurement error for 20 different stereo pairs

As we said the reconstruction depends strongly on the location of the feature elements. Fig.6 shows the average percentage measure errors of a few image pairs for different configurations of stereo calibration. Red-bar refers to measurements with automatic detection of landmark cylinder features; white bar is the accuracy by using automatic detection + additional stereo matched points (user selected). Blue-bar refers to manual selection of cylinders (light-green-bar corresponds to manual selection + stereo matched points). Overall the accuracy is better than 98%; only a few times the

measurement error is higher than 2 %. Although this is twice that of a tape, it has been proved sufficient in most practical situations. Clearly the accuracy of the measures is very accurate around the landmark and decreases for points far from it. There is almost no difference between the automatic location of cylinders and the manual location. In general the inclusion in stereo calibration of stereo pairs manually selected on the images does not improve the result significantly. However it is useful when the localization of the cylinders cannot be accurate enough.

5 Conclusions

The paper describes an industrial application of Landmark-based Stereo-Vision by using the vanishing point representation. To achieve a robust implementation our approach relies on the presence of an artificial landmark having an orthogonal triangular shape with sides of one meter length. Reflecting cylinders are used to allow an easy detection and location of landmark features, since both stereo images are acquired using a flash light (night and day conditions). The proposed systems is currently tested for the reconstruction of car-accident scenes by police operators, using a standard un-calibrated photo-camera, without any additional constraint for the user. The application software provides a simple and user-friendly interface to perform all image analysis and 3D measurement in a back-office environment. The achieved performance shows a measurement error that is always better than 2 % (with a 4Megapixel camera). Further research activity is in progress to achieve significant improvements in self-calibration and in multi-view integration.

Acknowledgements

Many thanks go to Biancamaria Zerbinati for her valuable support in the software development of the system integration environment and the Graphical User interface that has been greatly appreciated by the users for its effectiveness.

References

1. Real-Viz Image-Modeler, http://www.realviz.com/products/im/index.php
2. R.Cipolla, D.P.Robertson, E.G.Boyer, “Photobuilder-3 Models of Architectural Scenes from Uncalibrated Images”, proc. IEEE Int. Conf Multimedia Computing and Systems ICMCS’99, vol. 1, pp. 25-31, 1999.
3. B.Galvin, “Photogrammetry: fast crime-scene measuring with close-range photogrammetry”, Evidence Tech. Magazine, Sep.2004, www.EvidenceMagazine.com.
4. G.Garibotto, M.Corvi, C.Cibei, S.Sciarrino, “3D-MODS, Moving Obstacle Detection System”, Proc. Int. Conf on Image Analysis & Processing, ICIAP’03, Mantova, 2003.
5. O.D.Faugeras, “Three-Dimensional Computer Vision, A geometrical viewpoint, MIT press, Cambridge, MA, 1996.
6. PHS.Torr, DW.Murray Int J. Computer Vision 24 271-300 (1997).
7. E.K.Bas, J.D.Crisman, “An easy to installa camera calibration for traffic monitoring”, Proc. Of IEEE Conf. On Intelligent Transportation Systems, 1997, pp. 362-366.
8. B.Caprile, V.Torre, “Using Vanishing Points for Camera Calibration”, Int.J.Computer Vision, vol. 4, n.2, pp. 127-140, 1990.

Rectification-Free Multibaseline Stereo for Non-ideal Configurations

Hongdong Li and Richard Hartley

Research School of Information Sciences and Engineering,
The Australian National University,
ASSeT, Canberra Research Labs, National ICT Australia

Abstract. SSSD-based linear multibaseline stereo is an efficient implementation of multi-camera stereo vision system. This efficiency, however, vitally relies on the ideal configuration of all cameras. For dealing with non-ideal configurations, conventional stereo rectification algorithms can be used, but the performances are often still not satisfactory. This paper proposes a new algorithm to process non-ideally configured multibaseline stereo system, which not only avoids the rectification procedure but also remains the efficiency of SSSD at the same time. This is fulfilled by using the idea of tensor transfer used in image-based-rendering area. In particular, the multibaseline stereo is reformulated as a novel-view-synthesis problem. We propose a new concept of tensor-transfer to generate novel views as well as compute the depth map.

1 Introduction

It is well recognized that using more than two cameras can significantly improve the performance of stereo vision system, thanks to the redundant information contained in the multiple images [1][4].

In a multibaseline stereovision system, the multiple cameras can be arranged in different ways. This paper focuses on the **linear multibaseline system** proposed originally by Okutumi and Kanade[1]. For such system, the ideal configuration is shown in figure-1(a), where N cameras are arranged in such a way that all camera centers are collinear and all optical axes are in parallel. For such ideal configuration a very simple SSSD (sum of sum of squared differences) computation, based on the fact that the corresponding pixels in multiple views have a same inverse depth $\frac{1}{Z}$, is used to obtain the depth map. The computation is rather efficient, but effectively improves the stereo matching qualities. For this reason it has been popularly applied by many realtime stereo applications, particularly in mobile robot navigation fields[4][2][7]. Another practical reason is that: such linear configuration is easier to be mounted on the roof of an autonomous vehicle than other configurations (e.g., a L-shaped) which are more space consuming.

The efficiency of this SSSD computation, in fact, comes from the ideal configurations of the multiple cameras. Only when the N camera centers are collinear and their optical axes in parallel can we simply sum the SSD curves up. However,

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 810–817, 2005.

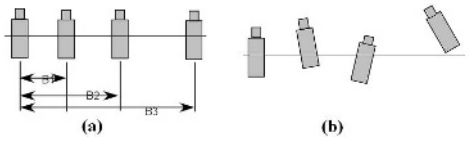


Fig. 1. Linear Multibaseline Stereo System: (a)ideal configurations; (b) non-ideal configurations

in many practical situations it is rather hard to build or maintain the ideality of the configurations. The insufficient precision in mechanization and assembly is one major reason. For example, it is rather hard to simultaneously adjust multiple cameras to make the optical centers precisely collinear and the optical axes parallel. Vibrations during the running of the system also affect the ideality of the camera configurations.

For dealing with generally configured multibaseline stereo, one may use traditional **rectification algorithms**[3][10][9]. However, the performances of these traditional rectification algorithms, when applying to the linear multibaseline system, are not satisfactory. Firstly, the computational efficiency is lost, since every camera needs to be rectified at least twice; Secondly, the matching performance is degraded significantly. The reason is that: every rectification involves image resampling and pixel interpolation, the introduced distortions will successively accumulate across images, which subsequently compromises the matching accuracy.

This paper proposes a new method to deal with the non-ideal linear multibaseline stereo (Figure-1(b)). Our method does not need a rectification processing. We borrow the idea of *image transfer* from image based rendering (IBR) area. By this idea, we re-formulate the multibaseline stereo matching problem as a typical novel-view-synthesis problem. We propose a new concept of inverse trifocal tensor by which a corresponding pixel-chain across multiple views is efficiently established.

Our method can be also used as a new novel-view-synthesis algorithm. Compared with existing methods, our method is featured by that it generates a very realistic novel-view image without resorting to any pre-computed precise depth map.

2 Previous Work

Most rectification algorithms were devised for binocular stereo[16][3]. So far, there exists only a very few papers that intently deal with multi-baseline rectification problem. Although it is commonly thought that one could trivially apply traditional binocular rectification algorithms to multi-baseline stereo in a pairwise manner, as we mentioned before this is not an efficient solution.

Mulligan and Daniilidis[4] proposed a method for processing non-parallel *trinocular* stereo. In their method, the image corresponding to the central camera is rectified twice. A large look-up-table is used to establish the correspond-

ing pixel link across multiple images. The computational overhead will become more serious when more cameras are used. Kimura and Kanade et.al introduced a rectification method also for three cameras[7]. There the three cameras are all rectified so that they are in an ideal orthogonal configuration. A transformation relationship is computed from the intersection points of mutual epipolar lines. However, their method does not work properly for multiple linearly configured cameras, because such configuration is a **degenerate case** of the epipolar relationship. Furthermore, the method can not be extended to more than three cameras. Paper[5] suggested a rectification method for general multibaseline configurations, but it requires prior knowledge of 3D positions of a set of feature points, which is therefore not suitable for many practical applications such as robot navigation.

Novel-View-Synthesis(NVS). The proposed multibaseline algorithm is based on the idea of Novel view synthesis. Using a small set of real input images to generate some novel images as if there were observed at some novel viewing positions, such is the process of novel-view-synthesis, an important application of computer vision. Methods of NVS can be roughly classified as two categories: 3D-reconstruction-based method, and image-based-rendering (IBR) method. Recently, the IBR methods have received more attentions than the reconstruction method, because they avoid much unnecessary and sometimes unstable explicit 3D reconstruction procedure, thus are more efficient.

3 Inverse Tensor Transfer

Trifocal tensor to three views is the same as fundamental matrix to two views. Here the tensor is a 3x3x3 data volume, just as the fundamental matrix is a 3x3 data array. Trifocal tensor encapsulates all projective geometric relationships among the three images.

For three images, denoted by image-1, image-2, and image-3, whose camera matrices are $\mathbf{P}_1$,$\mathbf{P}_2$ and $\mathbf{P}_3$, respectively. We denote the $\mathbf{P}_k^i$ for the i-th row of $\mathbf{P}_k$ matrix, and $\sim \mathbf{P}_k^i$ denote a sub-matrix of $\mathbf{P}_k$ by cancelling out the i-th row, then the trifocal tensor is computed as:

$$\mathcal{T}_i^{qr}\langle 123\rangle = (-1)^{i+1} \det \begin{bmatrix} \sim \mathbf{P_1}^i \\ \mathbf{P_2}^q \\ \mathbf{P_3}^r \end{bmatrix}, \tag{1}$$

where the subscriptions of $\mathcal{T}$ is called covariant index, the two superscriptions are called contra-variant indices, and the $\langle 123\rangle$ are camera indices.

For NVS application, the task is to synthesize a novel virtual image (denoted by image-0) from two given real image (for example, image-1 and image-2). Conventionally, the tensor transfer method is used to transfer a matched real pixel from image-1 (or image-2) to the virtual image-0. This is the traditional way of doing tensor transfer, where the tensor is computed from two real input images to the third virtual image. We call such tensor an *direct-tensor*, in order to distinguish it from our newly proposed *inverse-tensor*.

For three cameras, there are in total twelve linearly independent trilinear relations. However, only three independent trifocal tensors can be identified. The three tensors are distinguished by different choices of the first image. For example, for image-0, image-1 and image-2, totally there are only three independent tensors can be obtained, say, $\mathcal{T}\langle 012\rangle$, $\mathcal{T}\langle 120\rangle$ and $\mathcal{T}\langle 210\rangle$.

We give the definitions of inverse tensor transfer as: *an* ***inverse trifocal tensor*** *is a trifocal tensor transfer relationship* $\mathcal{T}\langle ijk\rangle$ *where the third camera index* k *is a real camera*. While, in contrast, *the conventional* ***direct trifocal tensor*** *is the transfer relationship from two real cameras to a third virtual camera*. Although there is no new geometric relationship introduced by this inverse tensor transfer, the motivation of especially conceptualize this is that it provides a simple mechanism to establish pixel correspondences across multiple real input images.

There are different ways of doing trifocal tensor transfer, which are distinguished by employing different incident relations. These incident relations include for example the point-line-line, point-line-point or point-point-point[11]. We use the point-line-point relation here in this paper, mainly for its computational simplicity.

4 Rectification-Free Multibaseline Algorithm

Our rectification-free multibaseline stereo algorithm proceeds step-by-step as follows:

1. First we specify the location of the virtual right-eye camera (image-0) with respect to the reference image (image-1) and compute the inverse tensors of $\mathcal{T}\langle 10k\rangle$, $k = 2, 3, 4, \cdots, N$ from the relative position of the cameras. (The two reference cameras' relative position is obtained from any relative orientation algorithm.)

2. We then apply the traditional window-based SSD (sum of square difference) stereo matching algorithm for the ideal stereo pair of image-1 and image-0, as if they are real images. The searching is limited within the disparity range of $[D_{min}, D_{max}]$. We assume that all the SSD values are zero, because actually there is no pixel values in the virtual image. Note that no *actual* pixel-similarity matching is performed here.

3. At every hypothesized disparity, we can determine the positions of currently matched pixels at all of the other real images, i.e., image-2, image-3,$\cdots$, using the corresponding inverse tensors, i.e., $\mathcal{T}\langle 102\rangle$, $\mathcal{T}\langle 103\rangle$,$\cdots$. The actual transferring scheme we use is the *point-line-point* formula which is given by: For $x_i \leftrightarrow x_j^{'}$, and $l_j^{'}$ passing through $x_j^{'}$, we have

$$x^{''k} = \hat{x}^i \cdot l_j^{'} \cdot \mathcal{T}_i^{jk}\langle 102\rangle \tag{2}$$

where the line is chosen to be the vertical line passing through the pixel of the virtual image. In fact, since image-1 and image-0 are in ideal stereo configuration, this line gives the optimal transfer relationship. In addition, the computational overhead is minimized.

4. Now for every disparity hypothesis, we have established a chain of matched pixels. Then we compute the conventional SSSD curve, by the formula:

$$SSSD(d) = \sum_{i=1}^{N} \sum_{(x,y)\in \mathsf{W}} \Phi(I_l^i(x+d,y) - I_r^i(x,y)) \tag{3}$$

where Φ is a robust-estimation kernel function to handle occlusions, w is SSSD window. Some candidates for the robust kernel Φ are:

$$\begin{aligned} \Phi(s) &= |s|, \\ \Phi(s) &= \tfrac{k^2}{2}\log(1+(s/k)^2) \end{aligned} \tag{4}$$

5. Choosing the minimal position as the resulting disparity, and output.

The inverse tensor plays a critical role in this algorithm because it offers a natural and simply way of relating N real corresponding pixels. The relations are guaranteed to be geometrically-valid since they are derived from the trifocal tensor. For continuous video sequence application or real-time robot navigation, the inverse tensor transfer relations need to be computed only once, and can be used later by making a look-up-table. So the computational overhead is not significant.

Our method also presents several contributions to the image based rendering area: Firstly, it alleviates a hidden problem which hinders most transfer based IBR methods, namely, the dense correspondence problem. Our method does not need any pre-computed depth map. We simply test for every possible disparity hypothesis whether all the matched pixels display a consistent color. In this regard, our method looks similar to the voxel-coloring algorithm[6] or image-based photo-hull methods. However, their methods perform in 3D scene domain, while ours fully performs in the image-domain, therefore is more efficient. Secondly, since we avoid dense correspondence, our method also frees from many troublesome problems that associate with the correspondence algorithms, for example, the *aperture problem*, highlight, and *textless regions* problem.

5 Experimental Results

Figure-2(1) gives the experimental setup of our multibaseline stereo system mounted on a prototype autonomous vehicles. The five cameras are arranged approximately linearly with optical axes in parallel. The central camera, which is chosen to be the reference camera, is also used as a time-base-synchronizer to ensure that all five cameras capture images at precisely the same time. The virtual camera is located to the right of the reference camera with a virtual baseline of 10×focal_length. Applying our new multibaseline algorithm, we obtained satisfactory 3D depth map and terrain reconstruction results. Figure-2(2) shows one example image, and figure-2(3) the obtained disparity map displayed in pseudo colors. Figure-2(4) is another example image, and figure-2(5) the corresponding terrain map. To compare the obtained depth map with the ground-truth depth

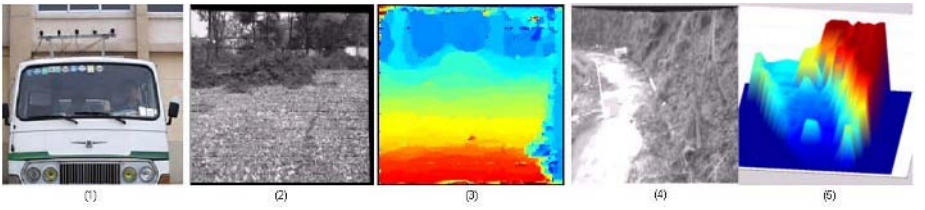


Fig. 2. A Multibaseline stereo system(read from left to right: (1)the camera system;(2) a sample image;(3) the computed depth map by our algorithm (4) another sample image (5) the computed terrain height map by our algorithm.)

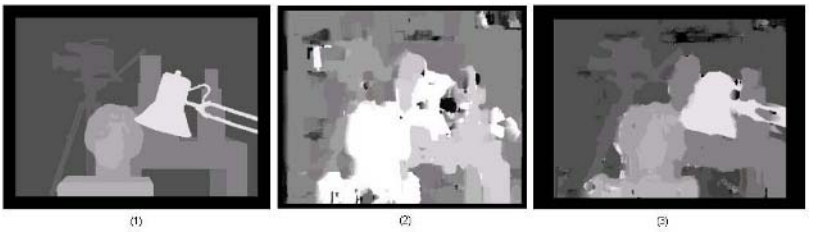


Fig. 3. Stereo correspondence results (read from left to right: (1) ground truth depth map; (2) result by conventional two-view SSD method; (3) result by our new multi-baseline method.)

map, we performed experiments on the Tsukuba stereo images. Prior to our algorithm, we intently introduce some arbitrary homographies to the original Tsukuba images (i.e., to make them to be un-rectified) in order to demonstrate our algorithm. The obtained depth map by our algorithm is shown in figure-3(3). For comparison purpose, figure-3(1) and 3(2) display the ground-truth depth map and the depth map obtained by a simple two-view SSD algorithm. It is seen that our algorithm produces good result at fairly efficient computation. We test our algorithm on more complex imageries. The camera is basically moving forward with a left side pan. So the images are in very general configurations, say, neither collinear nor parallel. In our experiments we do not use any calibration information. Even from the image contents itself, it is also a rather complex scenario for stereo matching: the white-wall and most part of the floor are all texture-less, and the ceiling light causes some intensity saturations at the right corner of the images. We have tested our algorithm on this sequence. The SSSD window size was set to 11x11. Figure-4(1) shows the obtained depth-map corresponding to frame-1. Though there contain some matching errors, the essential 3D scene structure is revealed. Remember that we actually started from a sequence which does not have any real stereo pair.

More interesting result is that: when we use this algorithm to generate novel views, even when the computed depth map is not so accurate, the synthesized image still looks very photo-realistic. Note the texture-less regions and highlight regions in the original image, which would destroy most conventional stereo

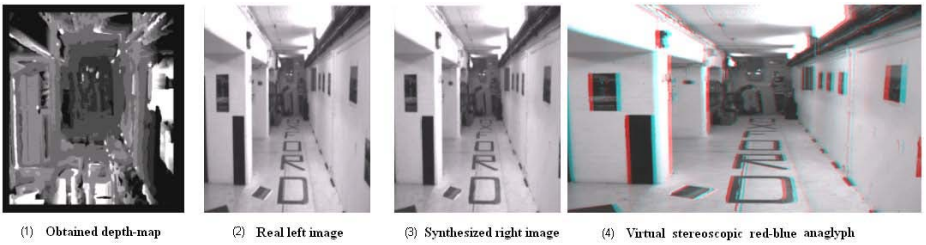


Fig. 4. The results by our virtual stereoscopic generation algorithm(read from left to right: (1) the obtained depth map;(2)the original input reference image; (3) the generated virtual right-eye image;(4) display the images of (2) and (3) using stereoscopic red-blue anaglyph.)

algorithms. Figure-4(2) and 4(3) show the generated stereo pair, where the left image is the reference image and the right image is the virtual right-eye image. We display them again in the red-blue anaglyph format in figure-4(4), from which the computed stereo disparities can be easily verified.

6 Conclusions

The purpose of stereo rectification is no other than establishing a simple indexing mechanism for stereo matching. This paper proposes an alternative method to accomplish this goal. Our method makes use of a new *inverse tensor transfer* technique. Although geometrically this inverse tensor follows the same idea of conventional direct trifocal tensor, it does provide a natural and easier way to retrieval corresponding pixels across multiple views.

Our currently SSSD computation is justified by the popularity of the approximate linear multi-baseline configurations. For more general configurations where for example the visual occlusions and projective distortions are apparent, we consider incorporating some recent techniques of wide-baseline matching or affine invariant matching(eg, SIFT, or [22][21]to improve the performance.

Acknowledgments

National ICT Australia is funded through the Australian Government's Backing Australia's Ability Initiative, in part through the Australian Research Council. The authors are grateful to the anonymous reviewers' valuable comments.

References

1. M. Okutomi and T. Kanade, A multiple baseline stereo. IEEE-trans on PAMI, Vol.15, No.4, pp.353-363, 1993.
2. T. Williamson, C. Thorpe, A Specialized Multibaseline Stereo Technique for Obstacle Detection, Proc IEEE-CVPR-98, pp.238-244, 1998.

3. C. Loop, Z.Y. Zhang, Computing rectifying homographies for stereo vision. Proc. IEEE-CVPR-99,v1, pp.125-131,1999.
4. J. Mulligan, K. Kaniilidis, Trinocular Stereo for Non-parallel Configurations, Proc. ICPR-2000, 1:567-570,2000.
5. T. Sato, M. Kanbara, N.Yokoya, H.Takemura, Dense 3-D Reconstruction of an Outdoor Scene by Hundreds-Baseline Stereo Using a Hand-Held Video Camera, Int. J. Comput. Vision, 47(1-3), pp.119-129, 2002.
6. S. M. Seitz and C. R. Dyer,Photorealistic scene reconstruction by voxel coloring. In Proc. IEEE-CVPR, pp. 1067-1073, 1997.
7. M.Kimura, H.Saito, T.Kanade, Stereo Matching between Three Images by Iterative Refinement in PVS, IEICE Trans. on Information and Systems, Vol.E86-D, No.1, pp.89-100, 2003.
8. Y. Li, S.Lin, H. Lu, S.B. Kang and H.Y. Shum, Multibaseline Stereo in the Presence of Specular Reflections. In Proc. ICPR-2002, 2002.
9. A. Fusiello, E. Trucco, and A. Verri, Rectification with unconstrained stereo geometry, Proc BMVC-1997, pp.400-409,1997.
10. J.M. Gluckman and S.K. Nayar, Rectifying Transformations That Minimize Resampling Effects, Proc. IEEE-CVPR-2001,2001.
11. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd edition, Cambridge University Press, 2004.
12. D. Papadimitriou and T. Dennis. Epipolar line estimation and rectification for stereo image pairs. IEEE trans on Image Processing, 5(4):672-676, 1996.
13. M. Pollefeys, R. Koch, and L. Gool. A simple and efficient rectification method for general motion. Proc. of the 7th ICCV-1999, 1999.
14. D.Scharstein and R.Szeliski, A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, IJCV 47(1-3), pp.7–42, 2002.
15. Y. Li, S. Lin, H. Lu, SB Kang, and HY Shum, Multibaseline stereo in the presence of specular reflections, ICPR-2002,, vol.3, pp. 573-576, 2002.
16. Lim, Mital, Davis and Paragios, Uncalibrated stereo rectification for automatic 3D surveillance, Proc. IEEE-ICIP-2004, 2004.
17. A. W. Fitzgibbon, Y. Wexler and A. Zisserman, Image-based rendering using image-based priors, Proc. ICCV-2003, 2003.
18. S. Avidan, A. Shashua. Novel View synthesis by Cascading Trilinear Tensors, IEEE-Trans on Visualization and Computer Graphics Vol.4, Iss.4 ,1998.
19. P. Mordohai, G. Medioni, Dense Multiple View Stereo with General Camera Placement using Tensor Voting, Proc. of 3DPVT'04, 2004.
20. J. Xiao, M. Shah, From Images to Video: View Morphing of Three Images. Vision, Modeling, and Visualization, 2003. (VMV2003),pp.495-502, 2003.
21. P. Pritchett, A.Zisserman, Wide baseline stereo matching, Proc. ICCV-1998,pp.754-760, 1998.
22. V. Ferrari, T.Tuytelaars, L. Van Gool, Wide-baseline muliple-view Correspondences, IEEE-CVPR-2003, pp711-718, 2003.

Optimal Parameter Estimation for MRF Stereo Matching

R. Gherardi, U. Castellani*, A. Fusiello, and V. Murino

Dipartimento di Informatica, Università di Verona,
Strada Le Grazie 15, 37134 Verona, Italy
Tel +39 045 802 7988, Fax +39 045 802 7068
umberto.castellani@univr.it

Abstract. This paper presents an optimisation technique to select automatically a set of control parameters for a Markov Random Field applied to stereo matching. The method is based on the Reactive Tabu Search strategy, and requires to define a suitable fitness function that measures the performance of the MRF stereo algorithm with a given parameters set. This approach have been made possible by the recent availability of ground-truth disparity maps. Experiments with synthetic and real images illustrate the approach.

1 Introduction

Three-dimensional (3D) reconstruction is a fundamental issue in Computer Vision, and in this context, structure from stereo plays a major role. The process of stereo reconstruction aims at recovering the 3D scene structure from a pair of images by searching for *conjugate points*, i.e., points in the left and right images that are projections of the same scene point. The difference between the positions of conjugate points is called *disparity*. The search is based on a matching process that estimates the "similarity" of points in the two images on the basis of local or punctual information [10]. A wide class of methods based on Markov Random Fields (MRF) models [8] has been recently introduced (see [10] for a review). Even if those methods have proved effective for the estimation of the stereo disparity, they often need a thoroughly phase of manual tuning of the free parameters that occurs in the MRF functional, using trial and errors.

In this paper we propose a technique capable of automatic selection of the "best" free parameters, based on an optimisation algorithm and a suitable fitness function that measures the performance of the MRF stereo algorithm with a given parameters set.

We consider the probabilistic stereo method R-SMW, proposed in [1], where the winner-takes-all approach of the Symmetric Multiple Window (SMW) algorithm [5] is relaxed by exploiting the non-determinism of the MRF. The MRF functional has two free parameters that in this paper are computed as the result of an optimisation based on the Reactive Tabu Search [6,2], which mitigates the

* Corresponding author.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 818–825, 2005.

problem of local minima trapping while driving the search to unexplored regions of the solution space. The fitness function is defined by comparing the output disparity with a known ground-truth.

Similar approaches, based on Genetic Algorithms, have been proposed in the past, focusing on different applications [3,9].

2 The R-SMW Method

The R-SMW algorithm [1] is a probabilistic relaxation of the SMW algorithm [5], using a Markov Random Filed.

In general, when an MRF model is applied to computer vision problems, the image is interpreted as a realization of a discrete stochastic process in which each pixel i is associated to a random variable D_i. By applying the Bayes rule and the Hammersley-Clifford theorem, the estimated solution (i.e., the solution that satisfies the Maximum A-Posteriori probability criterion) is obtained with the minimisation of $U(d \mid g) = U(g \mid d) + U(d)$, where $U(g \mid d)$ is the *observation model* and $U(d)$ is the *a-priori model*. In particular, the observation model describes the noise that degrade the image and the a-priori model describes the a-priori information independent from the observations, like, for instance the smoothness of the surfaces composing the scene objects. In the proposed application, in order to deal with the stereo problem [4], the scene is modelled as composed by a set of planes located at different distances to the observer, so that each disparity value corresponds to a plane in scene. Therefore, the a-priori model is *piecewise constant*.

Whilst the a-priori model impose a smoothness constraint on the solution, the observation model should describe how the observations are used to produce the solution. In the R-SMW method, the observation term encodes a multiple windows heuristic inspired from the SMW algorithm. For each site, we take into account all the disparity values in the neighbourhood weighted by the related SSD error values.

First a disparity map is computed using the simple SSD matching algorithm, taking in turn the left and the right images as the reference one. This produces two disparity map, which we will call left and right, respectively. A left-right consistency constraint is implemented by coupling the left disparity and the right disparity values.

In order to define the MRF model, we introduce two random fields D^l and D^r to estimate the left and the right disparity map, two random fields G^l and G^r to model the left and the right observed disparity map, and two random field S^l and S^r to model the SSD error. The field D^l (or equivalently D^r) will yield the output disparity.

In the following we shall describe the MRF functional, by defining the the a-priori model, the observation model, and the left-right consistency term[1].

[1] In the next two subsections, will shall omit superscript l and r in the field variables. It is understood that the a-priori model and the observation model applies to both left and right fields

2.1 A-Priori Model

With the a-priori term we encode the hypothesis that the surfaces in the scene are locally flat. Indeed, we employ a piecewise constant model, defined as:

$$U(d) = \sum_{i \in I} \sum_{j \in N_i} \delta(d_i, d_j) \tag{1}$$

where d_i and d_j are the estimate disparities value (the realization of the field D) and the function $\delta(x, y)$ is defined as:

$$\delta(x, y) = \begin{cases} 0 & if \quad x = y \\ 1 & otherwise \end{cases} \tag{2}$$

This term introduces a regularisation constraint, imposing that all pixels assume the same value in a region, thereby smoothing out isolated spikes.

2.2 Observation Model

In order to mimic the behaviour of the SMW algorithm, the observation model term introduces a local non-isotropic relaxation, favouring the neighbour observations with the lowest SSD value:

$$U(g, s \mid d) = \sum_{i \in I} \sum_{j \in N_i \cup \{i\}} \delta(d_i, g_j) \cdot \left(\frac{1}{s_j}\right) \tag{3}$$

where g is the observation disparity map (the realization of the field G), s is the observed SSD values (the realization of the field S), and d is the disparity estimate (the realization of the field D).

In this term, the estimate value at site i, d_i, is compared with all its observed neighbours $\{g_j\}_{j \in N_i}$ and with g_i. When d_i takes the disparity of one (or more) of its neighbours, one (or more) term(s) in the sum vanishes. The lower is the SSD error of the chosen disparity, the higher is the cost reduction.

2.3 Left-Right Consistency Constraint Term

Let d_i^l be the left disparity (i.e., the disparity computed taking the left image as the reference) at site i, and d_i^r the right disparity at site i. The left-right consistency constraint states that: $d_i^l = -d_{i+d_i^l}^r$. The corresponding energy term is:

$$V\left(d^l, d^r\right) = \sum_{i \in I} \delta\left(d_i^l, -d_{i+d_i^l}^r\right) \tag{4}$$

In this way we introduces a payload when the left-right constraint is violated.

2.4 Summing Up

The final MRF functional writes:

$$U\left(d_{\cdot}^{l}, d^{r} \mid g_{\cdot}^{l}, s_{\cdot}^{l}, g_{\cdot}^{r}, s^{r}\right) = k_1 \cdot \left[U\left(g_{\cdot}^{l}, s^{l} \mid d^{l}\right) + U\left(g_{\cdot}^{r}, s^{r} \mid d^{r}\right)\right] + \\ + k_2 \cdot \left[U\left(d^{l}\right) + U\left(d^{r}\right)\right] + k_3 \cdot V\left(d_{\cdot}^{l}, d^{r}\right) \tag{5}$$

where $U\left(g_{\cdot}^{l}, s^{l} \mid d^{l}\right)$ and $U\left(g_{\cdot}^{r}, s^{r} \mid d^{r}\right)$ are the observation model applied to the left and right disparity reconstruction, $U\left(d^{l}\right), U\left(d^{r}\right)$ are the a-priori models and $V\left(d_{\cdot}^{l}, d^{r}\right)$ is the left-right constraint term. The positive weights k_1, k_2, k_3 are the parameters that control the performance of the algorithms. As the absolute magnitude of the functional is not important in the MRF minimisation, we can set $k_1 + k_2 + k_3 = 1$, thereby reducing the free parameters to only two.

It is customary to adjust these parameters using trial and error; in the following of this paper we will describe our strategy for automatic optimal parameters selection based on Reactive Tabu Search [2].

3 Tabu Search

Tabu Search (TS) is a meta-heuristic introduced by [6] that systematically imposes constraints on the feasible solutions to permit exploration of otherwise forbidden regions of the search space. In particular, TS will not only remember the current and best solution but it will also keep information on the itinerary through the last solutions visited. Such information will be used in order to guide the transition from the current to the next solution.

The following components defines the TS.

Fitness function: this is a scalar function defined over the solution set, that return a score for each solution.

Move: a move is a procedure by which a new (feasible) solution is generated from the current one.

Neighbourhood: a neighbourhood of a solution is the set of all the solutions that can be reached with one move.

Tabu list: this is a list of moves that are forbidden (or tabu). Its length is fixed but it is updated dynamically with the last move that was picked.

Aspiration conditions: these are rules that overrides tabu restrictions. If the aspiration condition is satisfied, a tabu move becomes allowed.

The TS algorithm can be described as follows:

1. Given a starting solution, compute its fitness.
2. Generate the neighbourhood of the current solution, or, equivalently, a set of candidate moves. A move is allowed if it is not tabu or it satisfies the aspiration condition. Pick the allowed move that get to the best neighbouring solution and consider it to be the new current solution.
3. Repeat step 2 until some termination conditions are satisfied.

At each iteration, the chosen move is put in the tabu list, thereby preventing the algorithm to go back to recently visited solutions. However, given the fixed size of the tabu list, the search might be trapped in a cycle of length greater than the size list. In order to cope with this drawback, the *Reactive Tabu Search* (RTS) [2] has been proposed, which dynamically adjusts the tabu list size. In particular, the size is increased when configurations are repeated, otherwise it is reduced. Another reactive mechanism is the *escape*, which consist of a number of random step, and it is triggered whenever too many configurations are repeated too often. The reader is refereed to the bibliography [2,7] for more information on TS and RTS.

4 RTS Applied

The RTS is used to maximise a *fitness function* that measures the performance of the R-SMW stereo algorithm as the difference between the estimated disparity and the ground truth. The independent variables of the fitness function are the weights k_1 and k_2 (k_3 is recovered as $k_3 = 1 - k_1 - k_2$.) In more details the computation of the fitness function proceeds as follows:

- given a solution (parameter set) $s^{(\ell)} = (k_1^{(i)}, k_2^{(i)})$,
- run the stereo process with $s^{(\ell)}$ and find the disparity map D^ℓ,
- compute the fitness $f(s^{(\ell)}) = -\mathrm{err}(D^\ell, D^o)$, where D^o is the ground truth disparity.

Following [10], the disparity error is given by the fraction of wrong matches in non-occluded regions:

$$\mathrm{err}(D^\ell, D^o) = \frac{1}{N} \sum_{(i,j) \in I \setminus B} \delta \left(D^\ell(i,j), D^o(i,j) \right) \tag{6}$$

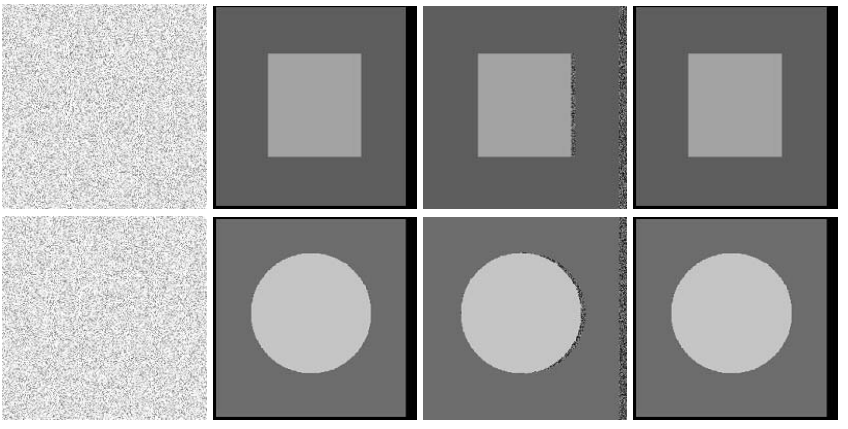
where N is the number of pixel, B is the set of occluded pixels (provided with the ground truth), and $\delta(x, y)$ has been defined in Equation (2).

A *solution* of the RTS is a point in the the region of the plane k_1, k_2 limited by the axes and by the line $k_2 = 1 - k_1$ (the search space). A *move* consists in changing the parameters value in such a way that the ratio between two of them is preserved. This gives three directions along which one can move starting from the current solution. A discrete change in the value of one parameter is obtained by flipping one bit of its binary representation. The *tabu list* is always updated with the last chosen move. The *aspiration condition* says that if a move leads to a better solution it is chosen even if is tabu.

5 Experiments

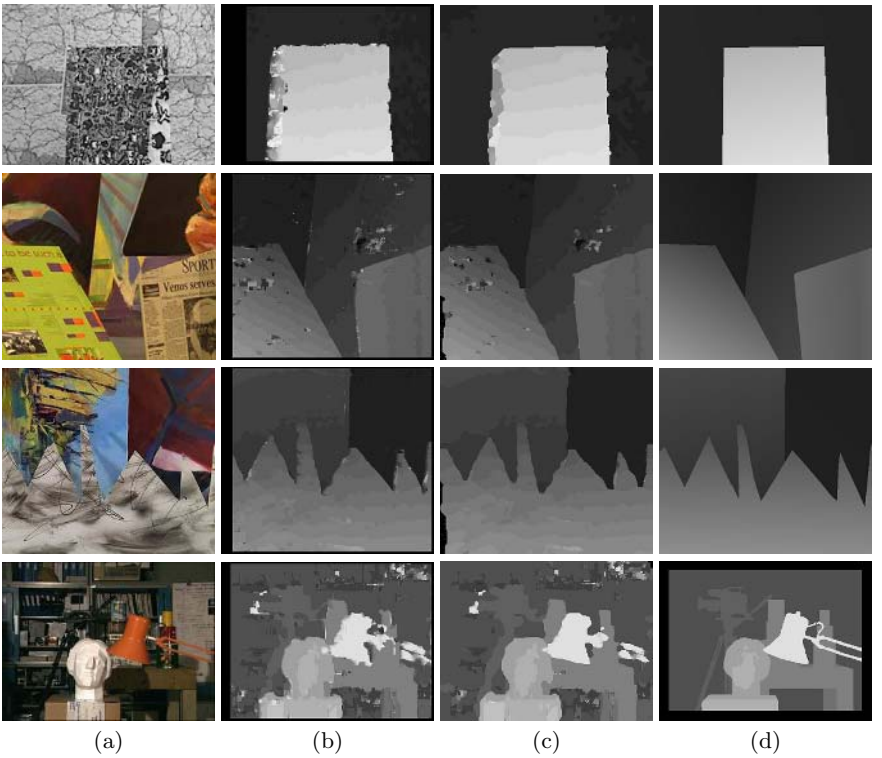
In this section experiments are reported for both synthetic and real cases.

First we estimated the optimal parameters for Random Dots Stereograms (RDS). The fitness function was computed using a square RDS and and circular



(a) (b) (c) (d)

Fig. 1. Random Dots Stereograms. (a) Left image; (b) disparity obtained by SMW; (c) disparity obtained by R-SMW with optimal parameters; (d) ground truth disparity.



(a) (b) (c) (d)

Fig. 2. Real experiments. (a) Left image of the stereo pair; (b) disparity obtained by SMW; (c) disparity obtained by R-SMW with optimal parameters; (d); ground truth disparity. Each row corresponds to a different stereo pair: (from top to bottom) Map, Venus, Sawtooth, and Tsukuba.

Table 1. Errors achieved by different parameters sets on different stereo pairs

Stereo pair	Single parameters	Err.	Joint parameters	Err.
Map	(0.03, 0.88, 0.09)	0.26	(0.68, 0.23, 0.09)	0.55
Venus	(0.67, 0.26, 0.07)	2.92	(0.68, 0.23, 0.09)	3.16
Sawtooth	(0.69, 0.21, 0.10)	2.38	(0.68, 0.23, 0.09)	2.41
Tsukuba	(0.56, 0.12, 0.32)	4.67	(0.68, 0.23, 0.09)	4.71

RDS (Figure 1). The RTS optimisation found the following values for the parameters $k_2 = 1, k_1 = k_3 = 0$, which reproduced the behaviour of the original SMW algorithm (not considering occluded areas), as can be seen in Figure 1. These values of the parameters make sense: the planar a-priori term is not needed since there is no noise; for the same reason and also because occlusions are not considered in the fitness function, the left-right consistency term is switched off.

Then we carried out experiments with the Middlebury data set [10], which is emerging as the de-facto standard data set for testing the performance of stereo algorithms. It consists of four stereo pairs: Map, Venus, Sawtooth and Tsukuba (Figure 2). The parameters estimation has been carried out using all the four stereo pairs (the global fitness function is the sum of the fitness for each set). The optimal parameters are reported in Table 1, in the column "Joint parameters". Figure 2 show the results obtained with these parameters for each stereo pair.

In order to assess the sensitivity of the parameters to the specific data set used for training, we estimated the optimal parameters separately for each stereo pair. The results are reported in Table 1, in the column "Single parameters". The error columns refer to the disparity error (i.e. the opposite of the fitness) value achieved by the given parameters set on a specific stereo pair.

It is worth noting that Sawtooth and Venus images are similar and so are the optimal parameters computed for these two stereo pairs. This seems to suggest that there are optimal parameters for classes of similar images.

6 Conclusion

The purpose of this paper has been to show that parameters tuning can be automated by using an optimisation strategy. We concentrated on stereo matching with a MRF-based algorithm (R-SMW) and used Reactive Tabu Search for parameters optimisation. The core ingredient is the fitness function, that measures the performance of a particular parameters set. The usefulness of such an approach is based on the claim that there are optimal parameters that are valid for classes of images, instead of being image-specific. Future work will aim at substantiating this claim.

Acknowledgments

This work was supported by the Italian Ministry of Research and Education under project LIMA3D (Low Cost 3D imaging and modelling automatic system).

References

1. V. Murino A. Fusiello, U. Castellani. Relaxing symmetric multiple windows stereo using markov randomfields. In A.K. Jain In M.Figureido, J.Zerubia., editor, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, number 2124 in Lecture Notes in Computer Science, pages 91–104. Springer, 2001.
2. R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6(2):126–140, 1994.
3. Luigi Cinque, Stefano Levialdi, Gianluca Pignalberi, Rita Cucchiara, and Stefano Martinz. Optimal range segmentation parameters through genetic algorithms. In *ICPR*, pages 1474–1477, 2000.
4. U. R. Dhond and J. K. Aggarwal. Structure from stereo – a review. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1489–1510, November/December 1989.
5. A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 858–863, Puerto Rico, June 1997. IEEE Computer Society Press.
6. F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
7. A. Hertz, E. Taillard, and D. de Werra. A tutorial on tabu search. Technical report, EPFL, Département de Mathétiques, MA-Ecublens, CH–1015, Lausanne, 1995.
8. S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Computer Science Workbench. Springer, Tokyo, 1995.
9. G. Pronzato and A.M. Wallace. Adaptive control of a boundary detection algorithm. In *Proc. of IEE Int. Conf. on Image Processing and its Applications,*, pages 356–360, Dublin, July 1997.
10. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondences algorithms. *International Journal of Computer Vision*, 1(47):7–42, 2002.

Dynamic Photometric Stereo

Melvyn Smith and Lyndon Smith

Machine Vision Laboratory,
Faculty of Computing, Engineering and Mathematical Sciences (CEMS),
University of the West of England, Bristol, BS16 1QY, UK
Telephone 0117 3283578
Melvyn.Smith@uwe.ac.uk
http://www.uwe.ac.uk/cems/research/melsmith/

Abstract. A new dynamic photometric stereo technique is presented, which facilitates the analysis of rapidly moving surfaces containing mixed two- and three-dimensional concomitant features. A new approach termed narrow infra-red photometric stereo (NIRPS) is described as an evolution of the existing photometric stereo (PS) technique. The method has application for the inspection of web materials and other moving surfaces considered difficult to analyse using conventional imaging techniques. Experimental results are presented in the paper.

1 Introduction

Photometric stereo (PS) can be used to recover both local surface orientation and albedo using three, or more, separate images acquired under differing directional illumination conditions, where the surface reflectance is normally assumed to be Lambertian and the lighting and viewing positions are distant. Previous work [1-4] has shown how PS can be used in surface topographic analysis through the recovery and analysis of a dense array of surface normals in the form of a surface bump map. Applications range from texture analysis to surface inspection [1]. In order to maintain correspondence between successive images it is necessary that the scene remain static between image acquisitions, hence the images are separated in time, where each image is acquired one after the other. This form of temporally multiplexed works well in static applications using an area array camera hence we may term this approach static PS (SPS). However in numerous industrial inspection applications it is often the case that the surfaces to be inspected tend to be moving and often moving very rapidly, i.e. >200m/minute. Under such circumstances, conventional image acquisition is normally performed using line-scan camera technology. It would be advantageous to be able to bring the benefits of PS to bear in on-line inspection of moving surfaces. However, achieving correspondence now becomes more problematic, as it may be impractical to capture three images. An alternative form of dynamic PS (DPS) is therefore proposed.

2 Conventional Static Photometric Stereo

The photometric stereo technique involves the capture of a number of views from a fixed spatial location while under differing carefully structured lighting configura-

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 826 – 833, 2005.

tions. By determining the normal at each pixel location, PS enables a full bump map to be obtained, which fully describes the surface topography. Smith [1, 4] has shown how the concept of a surface 'bump map', a term borrowed from computer graphics, could be used to represent a photometrically captured detailed 3D surface texture. In addition, it has also been shown how the bump map description offers opportunity for the characterisation of surface topographic features and textures in isolation from albedo through firstly mapping of the vector based data to a gradient space domain and then characterising the observed distribution [4]. Hence, by using what will be described here, as conventional, static or classical PS, precisely registered pixel based 2D albedo patterns and 3D surface topographic features and textures may be isolated and subsequently separately analysed [3]. A detailed explanation of PS will not be given here; instead the interested reader is referred to [1].

2.1 The Problem of Acquiring Multiple Images of Moving Surfaces

A limiting assumption forming the basis of SPS is that differing separate directional illumination configurations must be present in each of the acquired views of precisely the same region of the surface. While this is readily achieved in the static scenario, either by moving the light sources or by switching on and off differing lights between image acquisition stages, in terms of practical application allowing potential for rapid or uncontrolled relative movement between the camera and the surface, this assumption is found to be most restrictive. In many industrial inspection applications it is often the case that the component surfaces tend to be moving rapidly. For example, in the case of a web inspection this could be in excess of 200m/min. It is the ability to simultaneously acquire images under different lighting configurations that forms the basis of realising a dynamic photometric stereo (DPS) method.

3 Towards Dynamic Photometric Stereo

3.1 Spectral Multiplexing

It is useful to refer to each image / illumination configuration as a 'channel', where in order to minimise cross talk each channel must, as far as possible, be isolated from the others. Spectral multiplexing theoretically allows multiple images to be captured at a single point in space and time, where channel isolation may be realised in terms of light frequency separation. Two forms of spectral multiplexing present themselves, broad- and narrow-band.

Broadband colour photometric stereo. A number of related approaches have been described under what has been termed colour photometric stereo (CPS) or shape from colour, where colour channel separation is of the order of 100nm. As with much of the material relating to PS, CPS has generally been aimed at global shape recovery, rather than surface analysis per se. Nonetheless, CPS techniques are relevant to a discussion of the dynamic application of PS for surface analysis.

Drew [5] presents a method for recovering the shape of a Lambertian surface with unknown but uniform reflectance from a single composite RGB image. The method utilises a light that varies spectrally with direction and depends upon a linear

relationship between the RGB sensor response and the surface normal direction. Effectively a colour Lambertian model is used, where albedo is replaced by a RGB vector giving the colour of the illuminate reflected by the surface. Drew et al. later extend the method to curved surfaces with uniformly coloured patches, however, the method described is limited to curved surfaces and will not work for flat planar patches. The assumption of uniform colour in their earlier work would seem to limit application. A somewhat similar approach is described by Yasushi [6] using only white matt objects. Barsky et al. [7] describe a method for classifying rough surfaces using colour and gradient information. Although colour images are used, the method reduces to the grey scale case and is not CPS in the sense described here. Tian et al. [8] similarly use a colour image to recover the shape of non-Lambertian surfaces. An extended light source with differing colour to that of the object is used so that specular and diffuse reflection components may be detected simultaneously. By using colour information, regions of dichromatic, termed hybrid in their paper, reflection are segmented from regions of diffuse reflection. Unfortunately, limitations are again placed on variation in object colour by assuming uniform colouring. Only monochromatic surfaces are considered. Christensen et al. [9] also describe a variation of the photometric stereo method, utilizing colour instead of grey scale images. Their colour photometric stereo method is really a refinement of grey scale photometric stereo, and has been used to recover the shape of a coloured object from two or more colour images under white illumination. However, when applied to a purely diffuse refection their method corresponds to grey scale photometric stereo of the same surface. As with the work of Barsky et al. this is effectively grey scale PS and not CPS.

It is useful to introduce a specific definition for CPS. This is necessary in order to distinguish it from those methods utilising multiple images and multiple lights of equal chromacity (where although colour data is captured, the method simply reduces to conventional grey scale PS), and other methods. Therefore the term CPS is used here to specifically refer to any method in which colour information is used in the recovery of surface shape or topography.

Methods that have generally utilised single RGB image acquisition and lighting that varies spectrally with direction are often limited to monochromatic surfaces. An interesting interpretation of the colour photometric stereo approach is described by Detlef et al. [10]. Here three illuminates, red, green and blue, two in a dark-field and one in a bright-field configuration, are used with a single colour line scan camera. The two dark-field illuminates are arranged in a symmetrical configuration about the camera axis. By measuring the difference between the two images, an estimation of surface slope or gradient is arrived at. However, as only two illuminates are used only one degree of surface orientation is recovered in this configuration. Hence, it is not possible to obtain a full bump map and this limits its potential for full 3D-feature description. As with other methods the approach would appear to be limited to monochromatic and also to nominally planar geometry.

Mahdavieh [11] identifies a general limitation of the CPS method. Assuming the RGB intensities at a given surface point are known, while the x, y and z component of the surface normal and the RGB components of the surface reflectance spectrum remain unknown, the number of unknowns will exceed the number of equations that may be formulated. As such, the problem cannot be solved without introducing additional constraints. Mahdavieh proposes two solutions: either assume that the colour of

the object is monochromatic and just as importantly known a priori or add an additional image, captured using a broad spectrum white light located close to the camera position. Unfortunately, because the white light will cause a coupling with the RGB lights it becomes necessary to also introduce some other form of multiplexing, i.e. spatial. An alternative approach is now introduced, termed narrow band PS it serves to alleviate those limitations associated with CPS.

Narrow band colour PS - a new approach. A number of key limitations have been shown to present themselves when attempting to use a broad band colour PS approach. Firstly, when deploying widely spaced channels of visible light, a coupling is found to exist between surface colour and surface gradient, in which it becomes impossible to determine whether an observed surface colour is due to an unknown arbitrary surface reflectance or whether it is due to unknown surface gradient. Secondly, in order to use a standard RGB colour camera, some 100nm must separate each colour channel. This means that a surface of fixed arbitrary colour will appear at differing intensities under each coloured illuminate (e.g. a red surface will exhibit low radiance under blue illumination and high radiance under red illumination). In previous work these problems have been overcome using a priori knowledge, for example, by assuming that the surface is monochromatic or by previously acquiring a separate colour image, later registered with the PS images. The alternative is to utilise narrow frequency channels that are closely spaced at around only 20nm intervals or less. This new approach will be termed narrow channel or narrow band PS.

Figure 1 shows a typical spectral response for a red specimen surface with the region of closely spaced channels shown in the IR region of the spectrum. Approaching medium to long wave IR, i.e. 1.4 to 10 microns, further reduces sensitivity to changes in surface colour, i.e. different colours become metameric to one another. Also, it is known that both CCD and particularly CMOS cameras have excellent response in the IR region of the spectrum. Hence, using closely spaced channels within the infrared, surface colour data may be de-coupled from gradient data. In addition, if required an additional now de-coupled superimposed white channel (i.e. minus the IR) may be simultaneously included to provide fully registered colour data. This approach will be termed narrow infrared PS (NIRPS) [12]. A schematic configuration of the physical set-up is shown by Figure 2(a) with the detail of the camera optics shown in Figure 2(b).

The selection of appropriate filter optics becomes important in achieving good channel isolation and because of the limitations of real devices inevitably a compromise must be struck. For example, if the channel frequencies are spaced too widely then a given surface colour will appear with a different intensity within each channel causing variation in surface albedo to appear as topography in the bump map. If on the other hand the channels are too closely spaced then channel cross talk will be increased, distorting the recovered bump map by causing a shift in the observed position of the lights. In practice cross talk turns out to be less of an issue than albedo misinterpretation and can be reduced by calibration in which the contribution from the opposing two channels is subtracted from each channel.

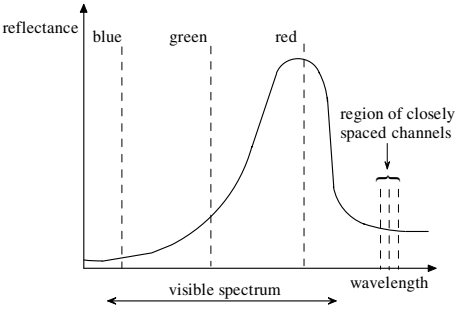
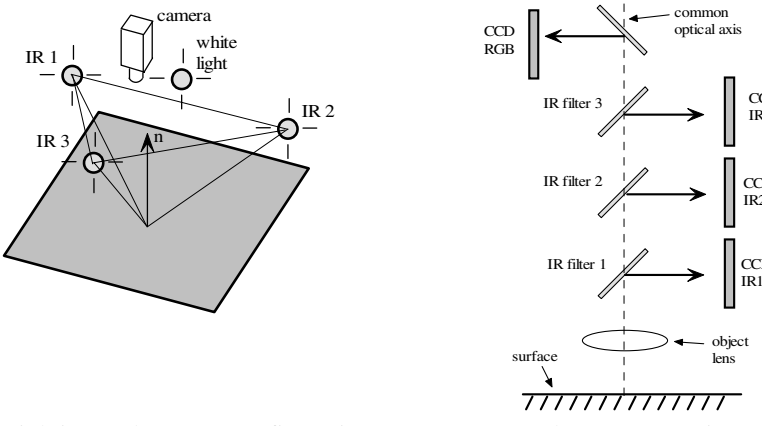


Fig. 1. Spectral distribution of a red surface



(a) Lighting and camera configuration (b) Camera optics

Fig. 2. Narrow IR photometric stereo (NIRPS)

3.2 Hybrid spatial / Frequency Multiplexing

A limitation of the NIRPS technique is the inherent complexity of camera optics, as shown by Figure 2(b). This may be considerably reduced by adopting a hybrid approach. Here images are isolated both in terms of spectral frequency and also by a close spatial displacement of one or two lines of pixels. A schematic configuration is shown in Figure 3.

This allows for considerable simplification of camera and optics. Instead of using multiple CCDs arranged off a single optical axis, adjacent CCDs (as shown) or adjacent regions of a single CCD may be used. As with spectral multiplexing, the various illuminates may be flooded into the inspection area, simplifying illumination in comparison with spatial multiplexing techniques, while channel separation takes place at the camera.

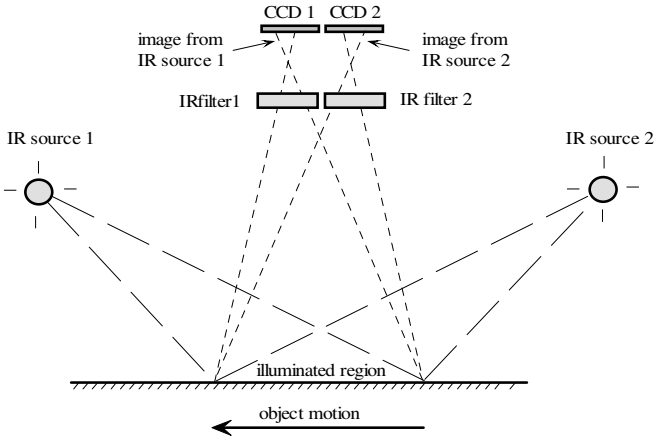
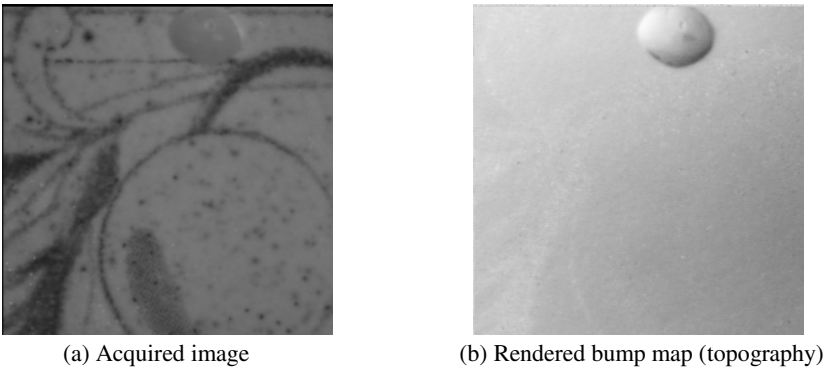


Fig. 3. Dynamic narrow infra-red photometric stereo (NIRPS)

4 Experimental Work

Static narrow IR photometric stereo (NIRPS). Initial static tests were performed using a ceramic surface exhibiting 3D topography with superimposed colouring. The surface includes a multi-coloured pattern and, at the top of the image, a three-dimensional feature, which also has a colour that differs from the rest of the sample. Three differing IR sources, 700, 710 and 720nm were used to simultaneously illuminate the scene in a directional manner, while images were taken using matched camera filters, one for each illuminate. Narrow band-pass interference filters were used for channel separation.



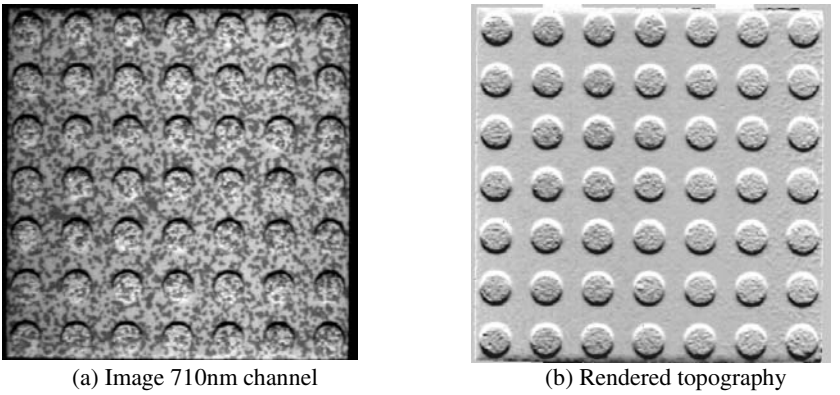
(a) Acquired image (b) Rendered bump map (topography)

Fig. 4. Narrow infra-red photometric stereo (NIRPS)

Figure 4 shows an albedo and a rendered bump map image. As can be seen, the surface topography has separated well in the rendered image, with only a faint rem-

nant of the albedo remaining. Cross talk between channels, observed by measuring the response of say the 750nm illuminate through the 730nm filter was found to be negligible.

Dynamic application using line scan imaging. Figure 5 shows the results of applying dynamic NIRPS using simultaneous illumination from IR LED line lights and line scan camera acquisition. The surface was travelling at 30m/minute.



(a) Image 710nm channel (b) Rendered topography

Fig. 5. Dynamic NIRPS operating at 30m/minute

As can be seen, the albedo is generally very well removed.

5 Conclusions

A new technique termed dynamic photometric stereo (DPS) has been described. DPS represents an evolution in the theory of photometric stereo by extending application to moving surfaces, thereby for the first time allowing application to numerous industrial inspection tasks. The technique has been implemented using a method known as narrow infrared photometric stereo (NIRPS). Experimental results have shown that NIRPS may be used to isolate concomitant two- and three-dimensional features on fast moving objects.

References

1. Smith, M. L., Surface Inspection Techniques - Using the integration of innovative machine vision and graphical modelling techniques, Professional Engineering Publishing, ISBN 1-86058-292-3, (2000).
2. Smith, M. L., Smith, L. N., Machine vision inspection, in Xipend Xu (ed), Machining of Natural Stone, Trans Tech Publishing, ISBN 0-87849-927-X, (2003).

3. Smith, M. L., Farooq, A. R., Smith, L. N., Midha, P. S., Surface texture inspecttion using conventional techniques applied to a photometrically acquired bump map, Sensor Review, Vol. 20, No. 4, (2000).
4. Smith, M. L., The analysis of surface texture using photometric stereo acquisition and gradient space domain mapping, *Image and Vision Computing*, Vol. 17, No. 14, (1999).
5. Drew, M. S., Shape from colour, Technical report, Centre for Systems Science/LCCR TR 92-07, School of Computing Science, Simon Fraser Uni, Vancouver, BC, (1992).
6. Yasushi, T., Shun'ichi, K., Tsunenori, H., A method for shape reconstruction of solid objects based on colour photometric stereo, IPSJ SIGNotes Computer Vision Abstract No. 080-16, (2001).
7. Barsky, S., Petrou, M., Classification of 3D rough surfaces using colour and gradient information recovered by colour photometric stereo, Proceedings of SPIE, 4553, pp. 10-19, (2001).
8. Tian, Y., Tsui, H. T., Extended light source model and reflectance map for non-Lambertian surfaces, Journal of Optical Society of America, Vol. 14, No. 2, pp. 397-404, (1997).
9. Christensen, P.H. and Shapiro, L. G., Three-dimensional shape from color photometric stereo, *International Journal of Computer Vision*, Vol.13, No. 2, pp. 213-227, (1994).
10. Detlef, P., (inventor), Method and apparatus for automatic inspection of moving surfaces, European patent application EP 0 898 163 A1, (1999).
11. Mahdavieh, Y., Investigation of a colour photometric stereo vision system, PhD disertation, Faculty of Technology, University of Manchester, (1984).
12. Smith, M. L., and Smith, L. N., (inventors), 'Infra-red photometric stereo', patent application WO03/012412A2, filed July 2001.

3D Database Population from Single Views of Surfaces of Revolution

C. Colombo, D. Comanducci, A. Del Bimbo, and F. Pernici

Dipartimento di Sistemi e Informatica,
Via Santa Marta 3, I-50139 Firenze, Italy
{colombo, comandu, delbimbo, pernici}@dsi.unifi.it

Abstract. Solids of revolution (vases, bottles, bells, ...), shortly SORs, are very common objects in man-made environments. We present a complete framework for 3D database population from a single image of a scene including a SOR. The system supports human intervention with automated procedures to obtain good estimates of the interest entities needed for 3D reconstruction. The system exploits the special geometry of the SOR to localize it within the image and to recover its shape. Applications for this system range from the preservation and classification of ancient vases to advanced graphics and multimedia.

1 Introduction

3D object models are growing in importance in several domains such as medicine, architecture, cultural heritage and so on. This is due to their increasing availability at affordable costs, and to the establishment of open standards for 3D data interchange (e.g. VRML, X3D). A direct consequence of this is the creation of 3D databases for storing and retrieval; for this purpose many different descriptions of 3D shape have been devised.

Based on the 3D shape representation used, various 3D object retrieval methods have been developed. These can be grouped into methods based on *spatial decomposition*, methods working on the *surface* of the object, *graph-based* methods, and *2D visual similarity* methods. Spatial decomposition methods [1][2] subdivide the volume object into elementary parts and compute some statistics of occurrence of features extracted from them (also in histogram form [3]). Surface-based methods [4] rely on the extraction of salient geometric features of the object: they can range from basic features as bounding box and axes of inertia to more sophisticated measurements such as moments and position of curvature salient points. In [5], the 3D problem is converted into an image indexing problem, representing the object shape by its curvature map. Graph-based approaches employ the object topology represented by a graph; thus, the similarity problem is reduced to a graph comparison problem. Finally, 2D visual similarity methods compare the appearance of the objects using image-based descriptions, typically derived from several views.

3D shape acquisition of a real object can be done through CAD (fully manual authoring) or through automatic techniques such as 3D laser scanner and

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 834–841, 2005.

imaged-based modeling. Laser scanners use interference frequencies to obtain a depth map of the object. Image-based modeling methods rely instead on camera calibration and can be subdivided into active and passive methods. Active methods employ structured light projected onto the scene. Though conceptually straightforward, structured-light scanners are cumbersome to build, and require expensive components. Computationally more challenging, yet less expensive ways to obtain 3D models by images are passive methods, e.g. classic triangulation [8], visual hulls [6], geometry scene constraints [7]. A central role is assumed here by self-calibration, using prior knowledge about scene structure [9] or camera motion [10].

To obtain a realistic 3D model, it is important to extract the texture on the object surface too. Basically, one can exploit the correspondence of selected points in 3D space with their images, or minimize the mismatch between the original object silhouette and the synthetic silhouette obtained by projecting the 3D object onto the image. Texture can also be used for retrieval purposes, either in combination with 3D shape or separately.

In this paper we propose a 3D acquisition system for SORs from single uncalibrated images, aimed at 3D object database population. The paper is structured as follows: in section 2, the system architecture is described; section 3 provides an insight into the way each module works. Finally, in section 4, results of experiments on real and synthetic images are reported.

2 System Architecture

As shown in Fig. 1, the system is composed by three main modules:

1. Segmentation of SOR-related image features;
2. 3D SOR shape recovery;
3. Texture extraction.

Module 1 looks for the dominant SOR within the input image and extracts in an automatic way its *interest curves*, namely, the "apparent contour" and at least two elliptical "imaged cross-sections." The module also estimates the parameters of the projective transformation characterizing the imaged SOR symmetry.

The interest curves and the imaged SOR symmetry are then exploited in Module 2 in order to perform camera self-calibration and SOR 3D shape reconstruction according to the theory developed in [11]. The particular nature of a SOR reduces the problem of 3D shape extraction to the recovery of the SOR *profile*—i.e., the planar curve generating the whole object by a rotation of 360 degrees around the symmetry axis. The profile can be effectively employed as a descriptor to perform SOR retrieval by shape similarity.

Module 3 exploits the output of Module 1 together with raw image data to extract the visible portion of the SOR texture. In such a way, a description of the object in terms of its photometric properties is obtained, which complements the geometric description obtained by Module 2.

Image retrieval from the database can then take place in terms of shape-related and/or texture-related queries.

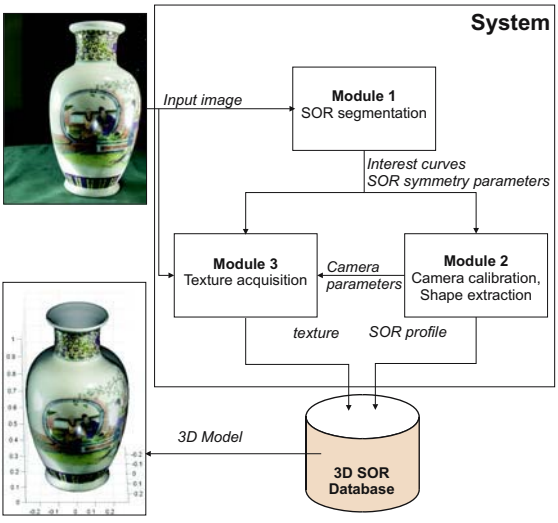


Fig. 1. System architecture

3 Modules and Algorithms

Module 1 employs the automatic SOR segmentation approach described in [12]. The approach is based on the fact that all interest curves are transformed into themselves by a single four degrees of freedom projective transformation, called *harmonic homology*, parameterized by an axis (2 dof) and a vertex (2 dof). The approach then consists in searching simultaneously for this harmonic homology and for the interest curves as the solution of an optimization problem involving edge points extracted from the image; this is done according to a multiresolution scheme. A first estimate of the homology is obtained by running the RANSAC algorithm at the lowest resolution level of a Gaussian pyramid, where the homology is well approximated by a simple axial symmetry (2 dof). New and better estimates of the full harmonic homology are then obtained by propagating the parameters through all the levels of the Gaussian pyramid, up to the original image (Fig. 2(a)). In particular, the homology and the interest curves consistent with it are computed from the edges of each level, by coupling an *Iterative Closest Point* algorithm (ICP) [13] with a graph-theoretic curve grouping strategy based on a Euclidean Minimum Spanning Tree [14]. Due to the presence of distractors, not all of the image curves thus obtained are actually interest curves; this calls for a curve pruning step exploiting a simple heuristics based on curve length and point density. Each of the putative interest curves obtained at the end of the multiresolution search is then classified into one of three classes: "apparent contour," "cross-section," and "clutter." The classification approach exploits the tangency condition between each imaged cross-section and the apparent contour.

As mentioned above, SOR 3D reconstruction (Module 2) is tantamount to finding the shape of its profile. This planar curve is obtained as the result of

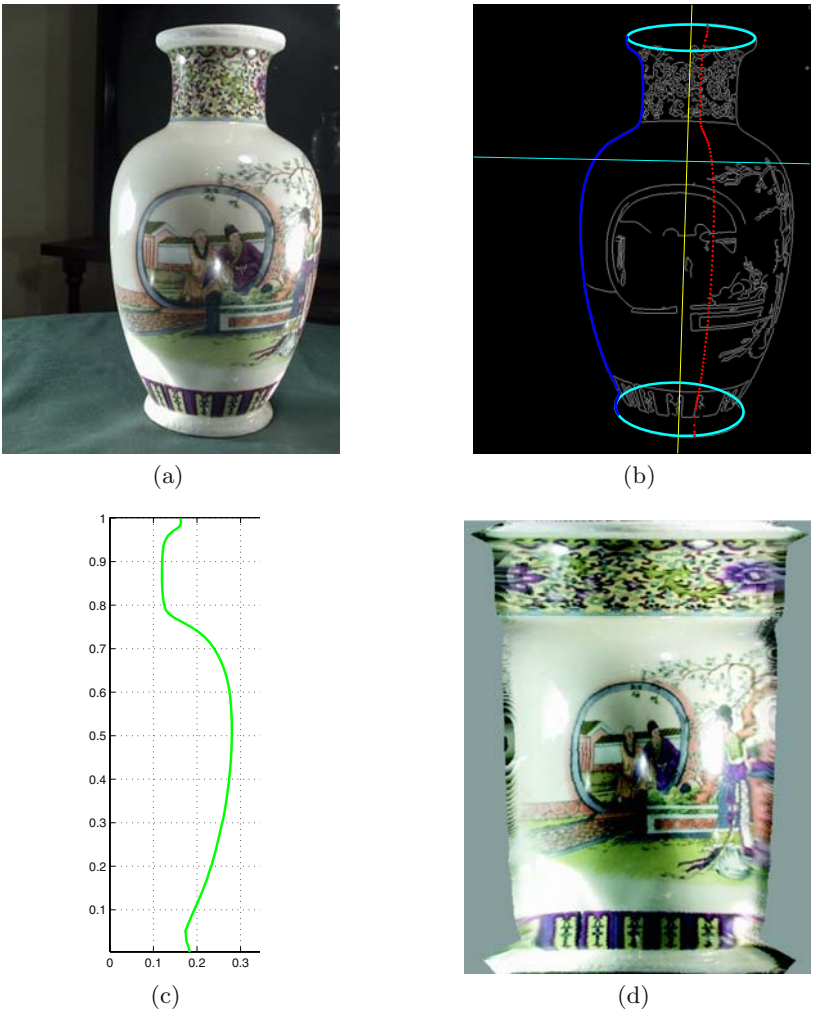


Fig. 2. From images to descriptions. (a): The input image. (b): The computed interest curves (one of the two halves of the apparent contour, two imaged cross-sections), the harmonic homology parameters (imaged SOR axis, vanishing line), and the recovered imaged meridian (dotted curve). (c): The reconstructed SOR profile. (d): The flattened texture acquired from the input image.

a two-step process: (*a*) transformation of the apparent contour into an imaged profile (see Fig. 2(b)); (*b*) elimination of the projective distortion by the rectification of the plane in space through the profile and the SOR symmetry axis (see Fig. 2(c)). Step *a* requires the knowledge of the homology axis and of the vanishing line of the planes orthogonal to the SOR symmetry axis; the latter is computed in Module 1 from the visible portions of at least two imaged

Table 1. System failures (automatic procedure) for a dataset of 77 images

type of system failure	%
bad edge extraction	24.68
harmonic homology not found	5.19
bad curve-grouping	6.49
bad ellipse extraction	12.99
total failures	49.35

cross-sections. Step *b* requires calibration information, also extracted from the homology and from the imaged cross-sections.

Texture acquisition is performed in Module 3. It exploits the calibration information obtained in Module 2, together with the vanishing line of the SOR cross-sections and the homology obtained in Module 1. The coordinates of the texture image are the usual cylindrical coordinates. Texture acquisition is performed by mapping each point in the texture image onto the corresponding point in the original image. This is equivalent to projecting all the visible SOR points onto a right cylinder coaxial with the SOR, and then unrolling the cylindrical surface onto the texture image plane. Fig. 2(d) shows the texture acquired from the original image of Fig. 2(a). Notice that, since the points that generate each half of the apparent contour generally do not lie on the same meridian, the flattened region with texture is not delimited with vertical lines.

4 Experimental Results

In order to assess the performance of the system, experiments have been conducted both on synthetic and real images. The former allowed us to use a ground truth and simulate in a controlled way the effect of noise and viewpoint on system percentage of success and accuracy. Experiments with real images helped us to gain an insight into system dependency on operating conditions and object typologies.

We indicate as "system failures" all the cases in which the system is unable to complete in a fully automatic way the process of model (shape, texture) extraction. System failures are mainly due to the impossibility for Module 1 to extract either the harmonic homology or the interest curves from raw image data: this can happen for several reasons, including large noise values, scarce edge visibility and poor foreground/background separation, specular reflections of glass and metallic materials, distractors as shadows and not-SOR dominant harmonic homologies, and the absence of at least two visible cross-sections. Tab. 1 shows the relative importance of the principal causes of system failures for a set of 77 images taken from the internet. The table indicates that most of the failures are due to the impossibility, in almost a quarter of the images, to obtain a sufficient number of edge points to correctly estimate the interest curves. The percentage of success for fully automatic model extraction is then slightly greater than 50%.

Table 2. Profile reconstruction error ε_r: mean value, standard deviation, min value, max value and number of failures; due to RANSAC initialization, the system has a random behavior also at zero noise level

σ	mean	std	min	max	failures
0	0.0010	0.0002	0.0008	0.0016	0
0.1	0.0011	0.0002	0.0008	0.0015	1
0.2	0.0013	0.0003	0.0009	0.0021	1
0.4	0.0014	0.0003	0.0010	0.0032	1
0.8	0.0021	0.0015	0.0011	0.0110	0
1.6	0.0039	0.0061	0.0013	0.0334	21

To cope with system failures, the system also includes a semi-automatic and a fully manual editing procedures. The former procedure involves only a slight intervention on the user part, and is aimed at recovering only from the edge extraction failures by manually filling the gaps in the portions of interest curves found automatically. Thanks to the semi-automatic procedure, system success increase to about 75% of the total, residual failures being due to bad homology estimates and misclassifications. The fully manual procedure involves the complete specification of the interest curves, and leads to a percentage of success of 100%. The three kinds of model extraction procedures give the user the opportunity to rank the input images into classes of relevance; in particular, models of the most relevant objects can in any case be extracted with the fully manual procedure.

If the system has completed its task on a given image (either with the automatic, semi-automatic, or manual procedure), it is reasonable then to measure the accuracy with which the shape of the profile is acquired. The profile reconstruction error is defined as

$$\varepsilon_r = \int_0^1 |\rho(t) - \hat{\rho}(t)| \, dt \ , \tag{1}$$

where $\rho(t)$ and $\hat{\rho}(t)$ are respectively the ground truth profile and the estimated one. Tab. 2 shows results using the ground truth profile

$$\rho(t) = \frac{1}{10}(\cos(\frac{\pi}{2}(\frac{19}{3}t + 1)) + 2) \tag{2}$$

of Fig. 3, for increasing values of Gaussian noise (50 Monte Carlo trials for each noise level $\sigma = 0, 0.1, 0.2, 0.4, 0.8, 1.6$).

If the error ε_r is regarded as the area of the region between $\rho(t)$ and $\hat{\rho}(t)$, we can compare it with the area given by the integral of $\rho(t)$ on the same domain: 0.1812. The error appears to be smaller by several orders of magnitude than this value, even at the highest noise levels. We can also notice how the number of failures increase abruptly for $\sigma = 1.6$; the failures are mainly due to a bad RANSAC output at the starting level.

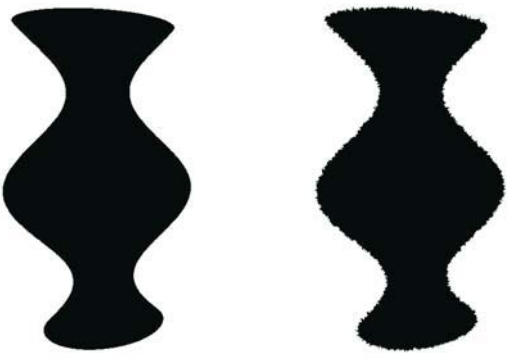


Fig. 3. Synthetic SOR views for $\sigma = 0$ (left) and $\sigma = 1,6$ (right)

5 Discussion and Conclusions

A 3D database population system has been presented, in which both shape and texture characteristics of SOR objects are acquired. Applications range from advanced graphics to content-based retrieval. Suitable representations of SOR models can be devised for retrieval purposes. Specifically, the SOR profile can be represented through its spline coefficients, and a measure of profile similarity such as that used in Eq. 1 can be used for shape-based retrieval. Texture-based retrieval can be carried out by standard image database techniques [15].

The experiments show that not all photos can be dealt with automatically, and that in half of the cases either a semi-automatic or a fully manual procedure have to be employed to carry out model extraction successfully. In order to obtain a fully automatic model extraction in any case, further research directions should include better edge extraction algorithms (color edges, automatic edge thresholding), a better grouping algorithm than EMST at the end of ICP, and some heuristics to discriminate whether a SOR is present in an image or not. In fact, EMST is good for outlier rejection inside ICP for its speed, but a better curve organization before curve classification can improve ellipse extraction; on the other hand, a SOR-discriminant heuristics can make the system able to work on a generic data-set of images.

References

1. Paquet, E., Rioux, M., A.Murching, Naveen, T., Tabatabai, A.: Description of shape information for 2D and 3D objects. Signal Processing: Image Communication **16** (2000) 103–122
2. Kazhdan, M., Funkhouser, T.: Harmonic 3D shape matching. In: SIGGRAPH '02: Technical Sketch. (2002)
3. Ankerst, M., Kastenmüller, G., Kriegel, H.P., Seidl, T.: 3D shape histograms for similarity search and classification in spatial databases. In: SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases (1999) 207–226

4. Zaharia, T., Prêteux, F.: Shape-based retrieval of 3d mesh-models. In: ICME '02: International Conference on Multimedia and Expo. (2002)
5. Assfalg, J., Del Bimbo, A., Pala, P.: Curvature maps for 3D Content-Based Retrieval. In: ICME '03: Proceedings of the International Conference on Multimedia and Expo. (2003)
6. Szeliski, R.: Rapid octree construction from image sequences. Computer Vision, Graphics, and Image Processing **58** (1993) 23–32
7. Mundy, J., Zisserman, A.: Repeated structures: Image correspondence constraints and ambiguity of 3D reconstruction. In Mundy, J., Zisserman, A., Forsyth, D., eds.: Applications of invariance in computer vision. Springer-Verlag (1994) 89–106
8. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2003)
9. Pollefeys, M.: Self-calibration and metric 3D reconstruction from uncalibrated image sequences. PhD thesis, K.U. Leuven (1999)
10. Jiang, G., Tsui, H., Quan, L., Zisserman, A.: Geometry of single axis motions using conic fitting. IEEE Transactions on Pattern Analysis and Machine Intelligence (**25**) (2003) 1343–1348
11. Colombo, C., Del Bimbo, A., Pernici, F.: Metric 3D reconstruction and texture acquisition of surfaces of revolution from a single uncalibrated view. IEEE Transactions on Pattern Analysis and Machine Intelligence (**27**) (2005) 99–114
12. Colombo, C., Comanducci, D., Del Bimbo, A., Pernici, F.: Accurate automatic localization of surfaces of revolution for self-calibration and metric reconstruction. In: IEEE CVPR Workshop on Perceptual Organization in Computer Vision (POCV) (2004)
13. Zhang, Z.Y.: Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision **13** (1994) 119–152
14. de Figueiredo, L.H., Gomes, J.: Computational morphology of curves. The Visual Computer **11** (1995) 105–112
15. Del Bimbo, A.: Visual Information Retrieval. Morgan Kaufmann (1999)

3D Face Modeling Based on Structured-Light Assisted Stereo Sensor

Boulbaba Ben Amor, Mohsen Ardabilian,
and Liming Chen

LIRIS Lab,
Lyon Research Center for Images and Intelligent Information Systems,
UMR 5205 CNRS, Centrale Lyon, France
{Boulbaba.Ben-amor, Mohsen.Ardabilian,
Liming.Chen}@ec-lyon.fr
http://liris.cnrs.fr/

Abstract. In this paper we present a 3D human face reconstruction framework based on stereo sensor coupled with a structured lighting source. Starting from two calibrated images, the active part (video projector) which project controlled lights, allows the operator to locate two sets of structured features with sub-pixel accuracy in both left and right images. Then, exploiting epipolar geometry improves the matching process by reducing its complexity from a bidirectional to a unidirectional search problem. Finally, we perform an adapted dynamic programming algorithm to obtain corresponding features in each conjugated scanline separately. Final three dimensional face models are achieved by a pipeline of four steps: (a) stereo triangulation, (b) data interpolation based on cubic spline models, (c) Delaunay triangulation-based meshing, and (d) texture mapping process.

1 Introduction

Over recent years, face detection [1], analysis, measurement and description have been applied widely in several applications [2] such as recognition, surveillance, multimedia document description, etc. Most face recognition technologies have two challenges: first, significant changes in lighting conditions cause system performance degradation. The alternative problem is their sensitivity to pose variations, indeed existing software compares probe image to gallery images taken at various angles. In our work, we propose to capture the 3D facial geometry in order to enhance recognition process. Having 3D images of faces we can generate a synthetic view at any angle and perform recognition algorithms.

In this paper we present a face reconstruction method based on a hybrid structured light and stereo sensor technique. This report is organized as follows: in section 2, we describe some existing methods for geometry recovering based on optical methods. Section 3 presents an overview of our system. A comprehensive description is given in sections 4, 5, 6, and 7. Finally, we present some experimental results and some future work in sections 8 and 9.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 842 – 849, 2005.

2 Related Work

In recent years, 3D human face acquisition technologies have made great progress. The successful solution of this problem has immense potential for applications in many domains. In this section we briefly review some approaches to recovering 3D facial geometry. Four potential optical methods are proposed: laser scanning, coded light range digitizers, silhouette-based methods, and multi-image/motion based approaches. Commercial 3D digitizers such as Cyberware [3] and Minolta [4] non-contact scanners are based on laser triangulation; Laser rays coming out of light source hit the object surface and are captured by a camera in a different angle using a rotating mirror. These devices take a short time to capture highly accurate reconstructions. However, they are expensive and the data are usually noisy requiring manual editing. In the case of multi-image based methods various approaches are developed. The classical ones use stereo sensors to acquire simultaneously two [5] or a set of images [6, 7], 3D information is found by triangulation; the intersection of optical rays going from projection centers joining corresponding features in images. The greatest challenge of these approaches is the matching process and the accuracy of the reconstructed models depends on precision of the matching. In the second category of these methods, the data source is a video sequence of face and Structure From Motion (SFM) algorithm is the most used approach. In [8] authors use SFM algorithms which are enhanced with a generic model as an initial solution. Here the obtained shape represents an approximation of the real face. Another solution is given by a structured light-based approach in which some kind of special lighting is directed to the object to be scanned. These active approaches help to solve the correspondence problem, which is a difficult task in passive methods. Depth information will be extracted from pattern deformation such as techniques presented in [9] and [10] but these devices are restrictive, so need to regroup a set of partial models. Other solutions have been developed such as silhouette extraction based methods [11], photometric methods [12] and face from orthogonal views methods [13] which produce an approximation of a real face.

3 Proposed Approach: Overview

Our approach is based on a binocular sensor, the minimal system for performing optical triangulation, coupled with a video projector which helps to resolve the matching process with a sub-pixel precision. Having a pair of points, each of them in a different image, which are projections of the same 3D point, this spatial point can be reconstructed using the cameras' light-rays intersection. But, the real challenge is how to establish corresponding pairs of points. Figure1 illustrates the overview of our system: in order to find camera parameters and the relationship between them, we calibrate the binocular sensor. After the calibration process, the correspondence problem, initially a two-directional search problem becomes a one-dimensional search problem, by applying image rectification to the pair of images. In the image acquisition process we take two images from each camera corresponding to normal and inverse light pattern projections. Consequently, the stripe intersections improve the edge detection process with a sub-pixel precision. Then an adapted dynamic programming algorithm is applied to perform matching feature sequences in left and right epipolar lines separately.

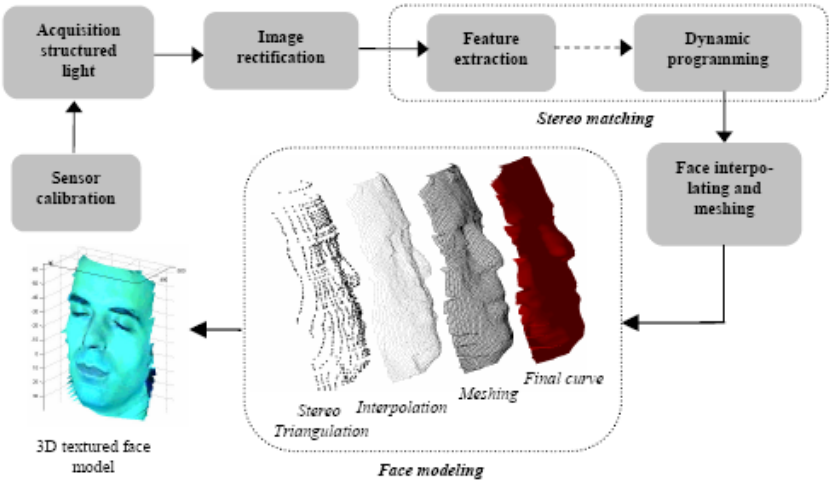


Fig. 1. Pipeline of 3d face modeling proposed approach

The final stage is to find a triangulation result for each pair of matching points. In order to obtain the 3D face model, we proceed by filtering, interpolating, meshing points and finally mapping texture onto a shape of face.

4 Calibration Step

This section briefly describes camera model geometry, stereo sensor geometry and the calibration processes which include computation of both intrinsic and extrinsic parameters. The main idea in camera calibration theory is to find a relationship between the 3D points of the scene and their 2D projecting point in the plane image. Here, the spatial point passes by different transformations in the camera acquisition process. Otherwise, calibration process computes changes from the initial 3D point P_w to the 2D image point P_i. First, the space point P_w is projected on an image plane in P_u by perspective projection. Second, the lens distortion changes the point position from P_u to P_d. Finally, P_d coordinates must be expressed in the image coordinate system to obtain P_I. The camera calibration technique used in our approach is described by Zhang in [14] where a planner pattern is used. Author computes a *closed-form solution* for initialisation, followed by a nonlinear optimization technique based on the maximum likelihood criterion which includes lens distortion.

After calibrations of each camera separately, we must compute the rigid transformation $A=(R,T)$ between them. Having this calibration data, it makes it possible to compute epipolar geometry and perform a rectification process in order to reduce the complexity of the correspondence problem. Indeed, given a feature point m in the left image, the corresponding feature point must lie on the corresponding epipolar line. In a standard stereo setup, conjugated epipolar lines are collinear and parallel to one of the image axis, usually the horizontal one. If the pair of images is taken with a general stereo configuration, an operation known as rectification can be applied to bring the two retinal planes to be coplanar to a common plane in space. The standard rectifica-

tion consists of selecting a plane parallel with the baseline $(O_L O_R)$. The two images are then re-projected onto this plane. The new images satisfy the standard stereo setup. After image rectification the epipolar lines become collinear in the rectified cameras and the correspondence finding is limited to a conjugate scanline.

5 Feature Extraction and Matching

The key step in the stereo approaches is the matching process which consists of finding the corresponding points between the left and right images. In our approach we propose to match a set of features which are detected by projecting negative and positive patterns of light on a subject. The projection of this kind of structured light helps to discriminate some feature points with sub-pixel precision and therefore increases precision of the matching process. The section below details the method used for stripe edge localization.

5.1 Stripe Edge Localization

The classical method for edge detection consists of detecting edges by simple binarization; i.e. with *pixel accuracy*. However, it is desirable to determine the stripe edge position with *subpixel accuracy*. This can be done by finding zero-crossings of the second derivative of the original image in the direction orthogonal to the stripes. Another method consists of locating intersections after projecting successively normal and inverse patterns.

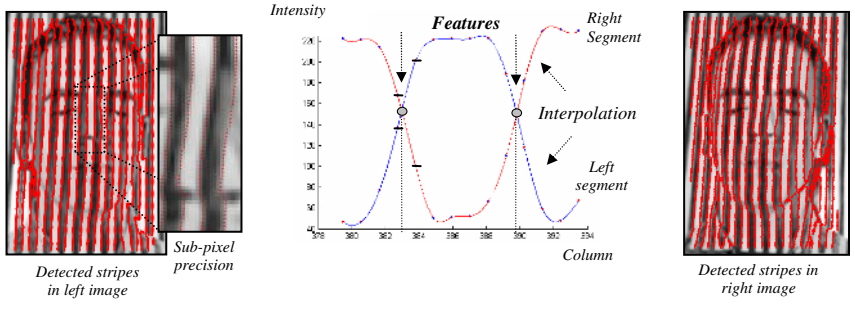


Fig. 2. Stripe boundary localization and results (detected features stereo images)

In figure 2 it has been shown how stripe edge can be detected by locating intersections of interpolated segments *AB* and *EF*. It shows also the results of stripe boundary detection in both left and right images.

5.2 Stereo Matching Based on Adapted Dynamic Programming Algorithm

Dynamic programming is a very useful technique for sequence matching and alignments. It solves an N-stage decision process as N single-stage processes. It reduces

the computational complexity to the logarithm of the combinatorial problem. In our approach we use this mathematical method to match left and right features. When images are rectified dynamic programming allows us to find the optimal solution for each scanline separately. The monotonic ordering constraint allows the global cost function to be determined as the minimum cost path through a disparity space image. The cost of the optimal path is the sum of the costs of the partial paths obtained recursively (1). Occlusions are modeled by assigning a group of pixels in one image to a single pixel in the other image and penalizing the solution by an occlusion cost *occ.* The *score*(q_i,e_i) is a normalized correlation measurement between features q_i and e_i.

$$\sigma(\Phi^*_{j,i}) = \begin{cases} 0, \text{ if } \quad j = 0 \quad \text{and} \quad i = 0; \\ \max\begin{cases} \sigma(\Phi^*_{j-1,i-1}) + \text{score}(q_j, e_i) \\ \sigma(\Phi^*_{j-1,i}) + \text{occ} \\ \sigma(\Phi^*_{j,i-1}) + \text{occ} \end{cases}, \\ \text{otherwise} \end{cases} \tag{1}$$

We define cost function as a matrix where lines and columns are indexed by left and right features for each scanline. However the principal disadvantage of this method is the possibility that local error may be propagated along a scanline, corrupting other potentially good matches. In this stage of our work, we simply filter the final result and false matches are cancelled in order to solve this kind of problem.

6 Face Modeling

To obtain a 3D face model we firstly triangulate matched points by finding intersecting points in space of obtained optical rays. Secondly, we mesh points after interpolation based on cubic spline models. Finally we map texture onto the obtained shape in order to add realism. Detailed descriptions are given in the following paragraphs:

6.1 Stereo Triangulation

Once having obtained the correspondence pairs, the 3D face model, or to be more precise its depth, can be recovered. The depth is (approximately) inversely proportional to the disparity (2), which symbolizes the shift of the corresponding points between the two images. Consequently, assuming the rectification, the disparity between a matched pair of points from the left image I_L and the right image I_R, respectively $I_L(u, r)$ and $I_R(v, r)$ is defined by $d(u,r)= u\text{-}v$. Now, we concentrate on the recovery of the position of a single point, *P*, from its projections, *p* and *q*. The classical relationship between depth Z and disparity d is given by formula (2):

$$\frac{b + d(u,r)}{Z - f} = \frac{b}{Z} \tag{2}$$

The 3D coordinates of a point $P(X,Y,Z)$, which correspond to the matched pair $I_L(u, r)$ and $I_R(v, r)$, can be expressed as follows (3):

$$X = \frac{b.u}{d(u,r)}; \qquad Y = \frac{b.r}{d(u,r)}; \qquad Z = \frac{b.f}{d(u,r)} \tag{3}$$

6.2 Model Interpolation and Meshing

Knowing a set of 3D points we interpolate in order to ameliorate the model's resolution and draw the face curve. Many interpolating data methods exist such as, linear, polynomial, Lagrange, Hermit, spline, etc. In our approach we use cubic spline functions [15] which are a very popular model for interpolation. The interpolating function is made up of a sequence of cubic polynomials across each interval of the data set curves that meet at the given data points with continuous first and second derivatives. Cubic spline interpolation is significantly better than some other interpolation methods for relatively smooth data such as faces.

Once having a 3D interpolated data set ($S=\{p_1, p_2, \ldots, p_n\}$), we triangulate in order to obtain the meshing curve of the face. For that, we use the Delaunay triangulation and Voronoi diagram duality, approach amongst the most useful data structures of computational geometry. The main idea of this algorithm is based on the Voronoi diagram which partitions the plane into convex regions, one per point or site. Given the Voronoi diagram of a set of sites the Delaunay triangulation of those sites can be obtained as follows: Given a set S of n distinct points in R^2 Voronoi diagram is the partition of R^2 into n polyhedral regions $Vo(p)$, $p \in S$. Each region $Vo(p)$, called the *Voronoi cell* of p, is defined as the set of points in R^2 which are closer to p than to any other points in S, or more precisely, $\mathrm{Vo(p)} = \{x \in R^2 / \mathrm{dist}(x,p) \le \mathrm{dist}(x,q) \forall\, q \in S - p\}$, Where dist is the Euclidean distance. The convex hull $conv(nb(S,v))$ of the nearest neighbor set of a Voronoi vertex V is called the Delaunay cell of V. The Delaunay complex (or triangulation) of S is a partition of the convex hull $conv(S)$ into the Delaunay cells of Voronoi vertices together with their faces. This Delaunay triangulation applied to the obtained space points gives a 3D face model and the results are shown in figure 3.

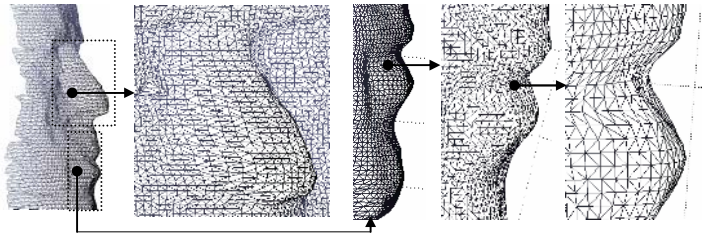


Fig. 3. Result of Delaunay triangulation process

6.3 Texture Mapping

In computer graphics, texture mapping refers to the technique where an image is pasted onto a three-dimensional surface. This technique can significantly increase the realism of a scene without increasing the complexity.

The final stage in our face reconstruction is to map texture onto the 3D shape in order to add realism to the face model. Currently, this step is performed by simply warping texture image to a 3D surface grid using interpolation. Figure 4 illustrates this process: first, we construct the texture image from the pair of left and right images then the warping procedure allows us to apply a texture onto the 3D shape with

Fig. 4. Texture mapping stage

specific parameters. This method is not the best one, so in our future work we will ameliorate image texture and provide some corresponding feature points from 3D shape and image texture.

7 Experimental Results

Figure 4 shows one textured model result. Our results are obtained from two images having 640×480 pixels where faces occupy 1/6 of the total surface and 16 stripes are projected onto the face subject. The presented model has 8081 vertices and 63948 triangles. However, these results can be enhanced by increasing the number of stripes and increasing photos' resolutions in order to cover more details of the face. Some parameters are adjustable such as *baseline*: the distance between the set of cameras and the distance from the subject to the sensor. At this moment, we are concentrating on computing the best set of parameters for reconstruction.

8 Conclusion and Future Work

We have presented in this paper a complete low-cost solution for 3D face acquisition using a stereo-structured-light coupled technique. The sensor is first calibred and parameters are extracted especially baseline and focal length. Second, epipolar geometry is also computed in order to reduce the complexity of the correspondence search problem. Then, the projection of normal and inverse structured light provides a set of point pairs with sub-pixel precision. The global matching optimization is performed by a dynamic programming algorithm for each scanline independently. Finally, depth is obtained by light-ray intersections. The final face geometry is achieved by performing interpolation and meshing techniques. The next step in our work is to introduce, in addition to the intra-scaline solution, the inter-scanline consistency constraint in order to enhance the matching process and reduce false matching.

References

1. D. Tsishkou, L. Chen and E. Bovbel, Semi-Automatic Face Segmentation for Face Detection in Video, Proc. of MEDIANET'04, Tozeur, Tunisia, pp. 107-118.
2. Kun Peng, Liming Chen and Su Ruan, Face Image Encryption and Reconstruction for Smart Cards Using Fourrier-Mellin Transform, Proc. of MEDIANET'04, Tunisia, pp 311-317.

3. Home page: www.cyberware.com
4. Home page: www.minolta-3d.com
5. E. Trucco and A. Verri, Introductory Techniques for 3D Computer Vision, ISBN 0-13-261108-2 Prentice Hall 1998.
6. D'Apuzzo N., Modeling human faces with multi-image photogrammetry. In: Corner, B.D., Pargas, R., Nurre, J.H. (Eds.), Three-Dimensional Image Capture and Applications V, Proc. of SPIE, Vol. 4661, San Jose, USA, 2002, pp. 191-197.
7. S. Lao, M. Kawade, Y. Sumi, F. Tomita: Building 3D Facial Models and Detecting Face Pose in 3D Space. 3DIM 1999: 390-404.
8. A.R. Chowdhury, R. Chellappa, S. Krishnamurthy, and T. Vu, 3D Face Reconstruction from Video Using a Generic Model, International Conference on Multimedia, Switzerland, pp. I:449-452, 2002.
9. E. Garcia, J.-L. Dugelay, H. Delingette, Low Cost 3D Face Acquisition and Modeling, ITCC, Las Vegas, Nevada, April 2001.
10. L. Zhang, B. Curless, and S. M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In Int. Symposium on 3D Data Processing Visualization and Transmission, Padova, Italy, June 2002.
11. Yang Liu, George Chen, Nelson Max, Christian Hofsetz, Peter McGuinness. Visual Hull Rendering with Multi-view Stereo. Journal of WSCG. Feb. 2004.
12. Chia-Yen Chen, Reinhard Klette and Chi-Fa Chen, 3D Reconstruction Using Shape from Photometric Stereo and Contours, October, 2003
13. Horace H S Ip and L.J. Yin, Constructing a 3D Head Model from Two Orthogonal Views, The Visual Computer, Vol 12, No. 5, pp. 254-266, 1996.
14. Z. Zhang, A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.
15. de Boor, C., *A* Practical Guide to Splines, Springer-Verlag, 1978.

Real-Time 3D Hand Shape Estimation Based on Inverse Kinematics and Physical Constraints

Ryuji Fujiki, Daisaku Arita, and Rin-ichiro Taniguchi

Department of Intelligent Systems, Kyushu University,
6-1, Kasuga-koen, Kasuga, Fukuoka, 816-8580, Japan
{fujiki, arita, rin}@limu.is.kyushu-u.ac.jp

Abstract. We are researching for real-time hand shape estimation, which we are going to apply to user interface and interactive applications. We have employed a computer vision approach, since unwired sensing provides restriction-free observation, or a natural way of sensing. The problem is that since a human hand has many joints, it has geometrically high degrees of freedom, which makes hand shape estimation difficult. For example, we have to deal with a self-occlusion problem and a large amount of computation. At the same time, a human hand has several physical constraints, i.e., each joint has a movable range and interdependence, which can potentially reduce the search space of hand shape estimation. This paper proposes a novel method to estimate 3D hand shapes in real-time by using shape features acquired from camera images and physical hand constraints heuristically introduced. We have made preliminary experiments using multiple cameras under uncomplicated background. We show experimental results in order to verify the effectiveness of our proposed method.

1 Introduction

Human hands, expressing our intension, are often used for communication. Therefore, hand shape recognition can be used in various interactive applications and user interface. We have developed hand shape estimation based on a computer vision approach, since unwired sensing provides restriction-free observation, or a natural way of sensing. There are, in principle, two approaches for hand shape estimation. One is classification of hand shapes into predefined categories based on pattern recognition techniques. The other is measurement of arbitrary hand shapes in 3D space. Though, the former approach can be used to indicate symbolic information, such as command labels of interaction, it can not present continuous information such as changes of hand shapes. On the other hand, the latter approach can acquire hand shape parameters in 3D space, or continuous information about 3D hand shape, and, therefore, it can be also applied to hand-manipulation-based control, such as control of robot hands, real-time 3D animation, etc.

Considering the applicability, we have adopted the latter approach. In this paper, we propose a novel method to estimate 3D hand shapes in real-time by using shape features acquired from camera images and physical hand constraints

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 850–858, 2005.

heuristically introduced. At first, we present the representation of a 3D hand model used our system. Then, we describe the details of hand shape estimation: extraction of hand features and 3D hand shape estimation by Inverse Kinematics with hand constraints. Finally, we show some experimental results in order to verify the effectiveness of our proposed method.

2 Related Works

The aim of our research is real-time estimation of 3D hand shape. Basically, there are two approaches proposed for 3D hand shape estimation.

- 2D appearance-based approach[1,2]
- 3D model-based approach[3,4]

The former is based on appearances of hands in 2D images, and essentially consists in a kind of template matching. Shimada et al.[1] represented variations of possible shape appearances as Locally-Compressed Feature Manifold (LCFM) in an appearance feature space. It is effective to prevent the system from tracking failures and to reduce the search area. Stenger[2] used a tree-based estimator based on Bayesian filter. This approach achieves coarse to fine search by approximating the posterior distribution at multiple resolutions, and hopeless sub-trees of the search space are not further evaluated.

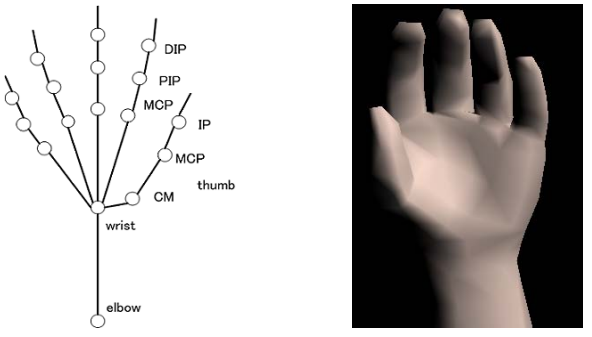
The latter is a method of extracting local hand features from images and estimating hand shapes, fitting a 3D hand model to the features. Ueda et al.[3] demonstrated the following method. Voxel representation is reconstructed from silhouette images by a multi-viewpoint camera system. Then a 3D hand shape is estimated using model fitting between a 3D hand model and the voxel model. Lu et al.[4] used a dynamic model and image force is calculated from image edges, optical flows and shading information. Then they perform fitting the dynamic model to image data applying image forces to the hand model.

Simply speaking, the former approach has the problem to deal with a large amount of templates. The latter has the problem of image feature missing by self-occlusion, since a human hand has many joints. Our approach is to extract the robust hand shape features from the images as much as possible even if it is not sufficient to know an exact hand shape. We estimate 3D hand shapes from the extracted features using hand constraints and geometric parameters of the hand model. In our system, based on the estimation result, we can re-extract effective image features by limiting a search range of image feature detection, and improve the estimation result.

Here, we use arcs on image contours obtained by image analysis, and we identify which finger (or fingertip) arc is detected. Then, we calculate the 3D positions of the arcs using multi-view analysis. Finally, we estimate hand shapes from the features by Inverse Kinematics with hand constraints.

3 Hand Model

In principle a human hand is a non-rigid object. In this paper, a human hand is approximated by a 3D rigid articulated object. This 3D hand model consists



(a) Skeleton Model (b) Skin Model

Fig. 1. Hand Model

of a skeleton model and a skin model(see Fig.1). The skeleton model consists of joints linked others, and has parent-child relation among the joints. Skin model is not currently used for hand shape estimation, except for visualization of results of hand shape estimation. In each finger, DIP (Distal Interphalangeal) and PIP (Proximal Interphalangeal) joints have 1 DOF (degree of freedom), and MCP (Metcarprophalangeal) has 2 DOFs. The wrist joint has 2 DOFs of yaw and pitch rotations. The elbow joint has 4 DOFs of translation and roll rotation, though, in this paper, the elbow is not fully estimated.

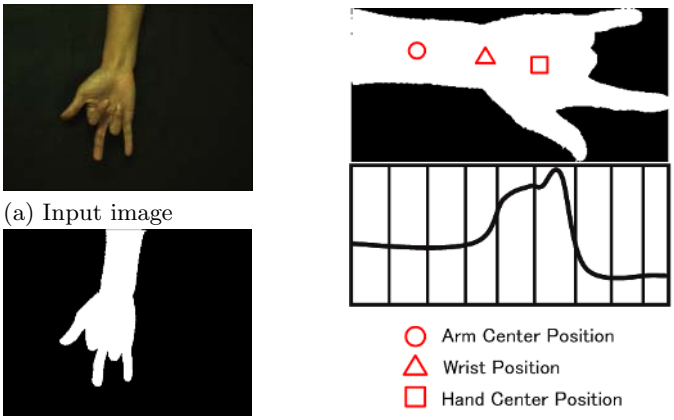
4 Feature Extraction of Hand

We extract shape features using image analysis. At first, we extract skin color region: we convert a color input image into a hue image (see Fig.2(a)) and, then, the image is smoothed for removing noises and binarized (see Fig.2(b).

4.1 Non-finger Feature Points

First, we extract non-finger shape features as follows (see Fig.2(c)).

1. Wrist Position
   We find a minimal-sized rectangle for extracted contour. This rectangle is normalized for rotation: x-axis is set to coincide with the arm direction. Generally, a hand is wider than an arm. Therefore we find the wrist position by searching for the minimum number of skin-color pixels projected on x-axis.
2. Arm Center Position
   The arm center position is computed as the centroid of skin-color region which is at the left part of the wrist position.
3. Hand Center Position
   The centroid usually dose not coincide with hand center, because it is strongly influenced by hand's opening and closing. We calculate an approximated



(a) Input image

(b) Skin color image

(c) Feature Points

Fig. 2. Image analysis

distance from every binary image pixel to the nearest contour pixel. Then the position of a pixel with the maximum value is judged to be the hand center position(x_{hand}, y_{hand})

4.2 Detection of Arcs on the Contour

As major features of 3D hand shape estimation, we detect arcs on the contour. These arcs are used as the positions of end effectors in inverse kinematics mentioned later.

1. Detection of Arcs on the Contour
   We detect arcs on the contour by curvature information, i.e., we detect contour points with large curvatures. The arcs correspond to joints projected outer-most in 2D image space. Here, we have to consider two problems: the correspondence between the arcs and fingers and correspondence between arcs extracted in two cameras. This problems is complex since we can make a lot of combinations. Therefore, we use physical hand constraints, the finger order and intervals between fingers, to reduce the combination.
2. Correspondence between arcs and fingers
   We decide which finger corresponds to each of arcs detected. When we detect five arcs, we can relate their order with the finger order. In other cases, i.e, when we can only detect k(< 5) arcs, we heuristically decide finger positions as follows.
   - We calculate possible finger position F_i (see Fig.3) based on the width of a hand detected and the hand model.
   - We define $C_j(c_{j,1}, \cdots, c_{j,k})$:$(j = 0, \cdots, \binom{5}{k})$ as a finger set which consists of selected k pieces from F_i.
   - The best combination C_j is determined as $argmin_j \sum_{m=1}^{k} ||c_{j,m} - a_m||$ (m=0,$\cdots$,k), where a_m is the position of *base of protrusion* corresponding

to the m-th arc. The base of protrusion is detected by contour tracing as follows:

- Beginning from an arc point, contours of its both sides are traced at the same speed until a concave point is found in one of the two contours (see Fig.3).
- a_m is the average of the positions of two points which are reached by the contour tracing.

When we calculate the distance $||c_{j,m} - a_m||$, only y coordinates of the positions are used. This procedure is introduced to reduce the influence of finger abduction.

As we decided the best combination, we can estimate which finger could not be detected. In case that a finger is not detected, we estimate its positions by finding a point with local maximum value of curvature whose y coordinate is similar to F_i(the missed finger).

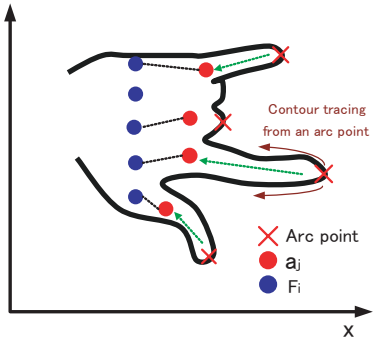


Fig. 3. Correspondence between arcs and fingers

4.3 Pitch and Yaw of Hand

We estimate pitch and yaw of a hand influenced by rotation of the wrist. They are easily calculated from the 3D positions of the arm center, the wrist position, and the hand center, which are defined in Fig.2.

5 Inverse Kinematics with Hand Constraints

5.1 Hand Constraints

A human's hand has many constraints[5,6,7]. Here, we consider hand constraints are combined with Inverse Kinematics.

1. Movable range of joints
   The movement of each finger is limited by movable ranges of joints. Movable range of each joint is shown Table 1.

Table 1. Movable range of joint

	DIP	PIP	MCP	abduction
Little	$0° \sim 70°$	$0° \sim 90°$	$-30° \sim 90°$	$-20° \sim 20°$
Ring	$0° \sim 70°$	$0° \sim 90°$	$-20° \sim 90°$	$-20° \sim 20°$
Middle	$0° \sim 70°$	$0° \sim 90°$	$-20° \sim 90°$	$-20° \sim 20°$
Index	$0° \sim 70°$	$0° \sim 90°$	$-40° \sim 90°$	$-20° \sim 20°$
Thumb	$-20° \sim 80°$	$-20° \sim 40°$	$-20° \sim 70°$	$-20° \sim 35°$

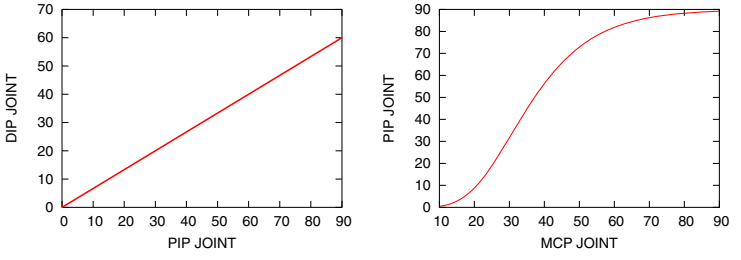


Fig. 4. Interdependence of Finger Joints: The figure above shows relation between PIP and DIP joint, the bottom shows relation between MCP and PIP joint

2. Limitation of abduction
   As we flex our fingers, adduction happens. Inversely, we extend our fingers, abduction happens. Considering this characteristics, we limit abduction of finger linearly with an MCP joint.
3. Interdependence of finger joints
   Each joint has a interdependence as follows:
   - On a grasping, a DIP joint has a linearity relationship with a PIP joint. ($\theta_{DIP} = \frac{2}{3}\theta_{PIP}$)
   - As an MCP joint bends, PIP and DIP joints bend slowly at first. Then their joints bend suddenly in the middle of the MCP joint flexion. Finally their joints bend again slowly. In brief, an MCP joint has correlation of an ess curve with PIP and DIP joints. However, a human hand can extend an MCP joint under PIP and DIP joints bended. Therefore we assume a PIP joint makes a constant angle, when an MCP joint angle is smaller than 10 degrees.

   On the basis of the above observation, we have heuristically introduced relational expressions shown in Fig.4.

5.2 Hand Shape Estimation Using IK

We estimate joint angles of fingers by Inverse Kinematics based on the hand constraints mentioned above. In general, IK is to determine joint angles $\theta_{\mathbf{i}}$ of a manipulator so that the position of an end effector, or a final point, $\mathbf{P}_n$, coincides with a given goal point $\mathbf{G}$: $\mathbf{P}_n(\theta_1, ..., \theta_n) = \mathbf{G}$: where the manipulator

has n segments. Here, the goal is given by an arc position detected and the target is a fingertip or a finger joint of the hand model.

We use Cyclic Coordinate Descent (CCD) method to solve IK[8]. The merit of the CCD method is that it estimates joint angles based on a posture in previous frame and it is a fast algorithm. This is effective for real-time processing.

Identification of Arcs. We decide correspondence between fingers (fingertip) and arc points. The algorithm is summarized as follows.
On a finger,

1. We set an arc as the goal, and fingertip as the target of IK.
2. If the distance between wrist and target is longer than the distance between wrist and parent joint, the target is refused. The joint must be on the outside on the wrist, because the arc detected is on the contour.
3. We solve Inverse Kinematics by CCD method, satisfying the hand constraints. Then, we calculate the error between the goal and the target.
4. Unless the target is an MCP joint, make the next target be the parent joint of the current target, and go to 2.
5. We select a joint which has the least error as the correct joint corresponding to the given arc.

Table 2. Processing time

Algorithm	Time (msec)
Convert image to hue color space	12
Detection skin color region	12
Extract hand features	20
IK calculation	14
Total	58

Estimation of Joint Angles. Even if we can identify an arc position correctly using the method above, angles of its child joints can not be estimated by Inverse Kinematics, except for the case that the arc position corresponds to a fingertip. We solve this problem by using interdependence of joint angles, and estimate those joint angles.

5.3 Post Processing

We can add the following processing so that we estimate more accurate hand shape. From the error of Inverse Kinematics, we can evaluate false estimation in image analysis. When the error of Inverse Kinematics estimation is pretty large, we suppose that the correspondences between the arc and the finger or the correspondence between two views for depth measurement is not correct. Then we estimate finger angles again by using another correspondence. In addition, we can know hand shape information from the result of Inverse Kinematics, and

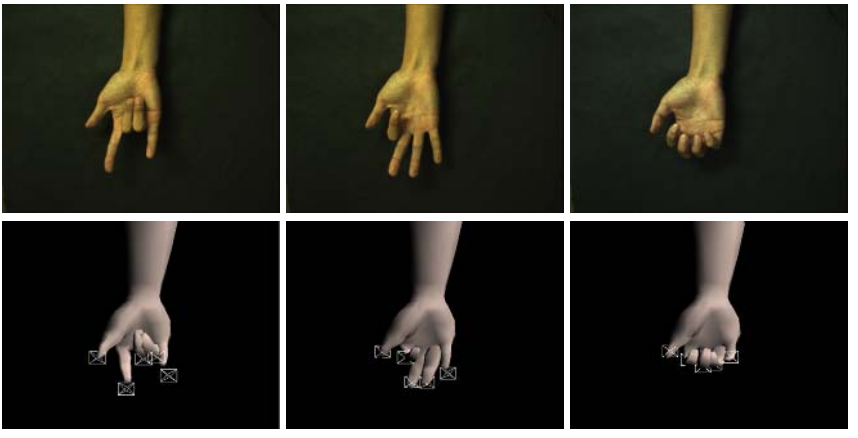


Fig. 5. The result of estimation

we have a chance to acquire effective image features such as fingertip position by limiting a search range of image feature detection. We can use those features for more accurate hand shape estimation.

6 Preliminary Experiment

We experimented the 3D hand shape estimation by our proposed method except for the post processing. In this experiment, we have used IEEE-1394-based color cameras (Point Grey Research Inc; Flea) with f:8 mm lenses, which are geometrically calibrated in advance. The images are captured with the size of 640×480 pixels. Several experimental results are shown in Fig.5, which indicates the effectiveness of our method. The processing time is shown Table. 2 using PC with Pentium *IV* (2GHz). It shows that our algorithm can be used for real-time applications.

7 Conclusion

In this paper, we have shown a real-time 3D hand shape estimation without special marker-sensors. The key point is that we use Inverse Kinematics with physical hand constraints in order to complement hand shape information not to be directly obtained image features. We have experimented our proposed method under non-complex background.

The next goal is to build a system which can handle the turn of the palm of the hand, by selecting two cameras which face to the front of a hand from multiple cameras. In other words, we estimate the roll rotation of elbow joint. We also have to extract the shape features which are effective to estimate more

accurate 3D hand shape from cluttered background. Acquisition of the geometrical parameters of 3D hand model, such as lengths of finger bones, from the first pose is also an important issue.

References

1. N.Shimada and Y.Shirai, "Hand Posture Estimation based on 2D Appearance Retrieval Using Monocular Camera," Proc. Int, Workshop on RATFG-RTS, pp.23-30, 2001.
2. B.Stenger, A.Thayananthan, P.H.S.Torr, and R.Cipolla, "Filtering using a tree-based estimater," Proc. ICCV, pp.1063-1070, 2003.
3. E.Ueda, Y.Matsumoto, M.Imai and T.Ogasawara. "Hand Pose Estimation for Vision-based Human Interface", IEEE Trans. on Industrial Electronics. Vol.50, No.4, pp.676-684, 2003.
4. S.Lu, D.Metaxas, D.Samaras and J.Oliensis, "Using multiple cues for hand tracking and model refinement," Proc. CVPR, pp.443-450, 2003.
5. C.Häger-Ross and M.H.Schieber, "Quantifying the Independence of Hand Finger Movements: Comparisons of Digit, Hands, nad Movement Frequencies," The Journal of Neuroscience, Vol.20, No.22, pp.8542-8550, 2000.
6. D.G.Kamper, E.G.Cruz, and M.P.Siegel, "Stereotypical fingertip trajectories during grasp," Journal of Neurophysiology, Vol.90, No.6, pp.3702-3710, 2003.
7. G.ElKoura, K.Singh "Handrix: Animating the Human Hand," Proc. SIGGRAPH, pp.110-119, 2003.
8. L.T.Wang and C.C.Chen, "A combined optimization method for solving the inverse kinematics problem of mechanical manipulators," IEEE Trans. on Robotics and Automations, Vol.17, No.4, pp.489-499, 1991.

Curvature Correlograms for Content Based Retrieval of 3D Objects

G. Antini, S. Berretti, A. Del Bimbo, and P. Pala

Dipartimento di Sistemi e Informatica,
Università degli Studi di Firenze,
via S.Marta 3, 50139 Firenze, Italy
{antini, berretti, delbimbo, pala}@dsi.unifi.it

Abstract. Along with images and videos, 3D models have raised a certain interest for a number of reasons, including advancements in 3D hardware and software technologies, their ever decreasing prices and increasing availability, affordable 3D authoring tools, and the establishment of open standards for 3D data interchange. The resulting proliferation of 3D models demands for tools supporting their effective and efficient management, including archival and retrieval.

In order to support effective retrieval by content of 3D objects and enable retrieval by object parts, information about local object structure should be combined with spatial information on object surface. In this paper, as a solution to this requirement, we present a method relying on curvature correlograms to perform description and retrieval by content of 3D objects.

Experimental results are presented both to show results of sample queries by content and to compare—in terms of precision/recall figures—the proposed solution to alternative techniques.

1 Introduction

Beside image and video databases, archives of 3D models have recently gained increasing attention for a number of reasons: advancements in 3D hardware and software technologies, their ever increasing availability at affordable costs, and the establishment of open standards for 3D data interchange (e.g. VRML, X3D).

Three-dimensional acquisition of a real-world object, capturing both object geometry and its visual features (surface color and texture), can be achieved through many different techniques, including CAD, 3D laser scanners, structured light systems and photogrammetry. Thanks to the availability of these technologies, 3D models are being created and employed in a wide range of application domains, including medicine, computer aided design and engineering, and cultural heritage.

In this framework the development of techniques to enable retrieval by content of 3D models assumes an ever increasing relevance. This is particularly the case in the fields of cultural heritage and historical relics, where there is a growing interest in solutions enabling preservation of relevant artworks (e.g. vases, sculptures, and handicrafts) as well as cataloguing and retrieval by content. In these fields, retrieval by content can be employed to detect commonalities between 3D objects (e.g. the "signature" of the artist) or to monitor the temporal evolution of a defect (e.g. the amount of bending for wooden tables).

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 859–866, 2005.

1.1 Previous Work

Methods addressing retrieval of 3D models can be distinguished based on different aspects, such as the type of representation used for geometry, the use of information about models' appearance (i.e. colour and/or texture), the need for manual annotation.

Generally, two broad classes of approaches can be distinguished: *view-based* and *structure-based*. In the former, salient features of an object are extracted from a set of 2D views of the object itself. In the latter, object features are computed directly in the three-dimensional space in order to capture prominent characteristics of the object structure.

Description and retrieval of 3D objects based on description and retrieval of 2D views has been addressed in [1] and [2]. However, the effectiveness of these solutions is limited to description and retrieval of simple objects. In fact, as complex objects are considered, occlusions prevent to capture distinguishing 3D features using 2D views.

Recently, a hybrid approach—that is not entirely view-based or structure-based—has been proposed in [9] relying on the use of spin images for content description and matching. Description of 3D structure for the purpose of recognition or retrieval has been addressed for some time. A few authors have investigated analytical 3D models, but this is not always a viable solution, as there are many limitations in providing parameterizations of arbitrary models. In [3] retrieval of 3D objects based on similarity of surface segments is addressed. Surface segments model potential docking sites of molecular structures.

Much attention has been recently devoted to free-form (i.e. polygonal) meshes. The system developed within the Nefertiti project supports retrieval of 3D models based on both geometry and appearance (i.e. colour and texture) [4]. Also Kolonias et al. have used dimensions of the bounding box (i.e. its aspect ratios) and a binary voxel-based representation of geometry [5]. They further relied on a third feature, namely a set of paths, outlining the shape (*model routes*). In [6] a method is proposed to select feature points which relies on the evaluation of Gaussian and median curvature maxima, as well as of torsion maxima on the surface. In [7], Elad et al. use moments (up to the 4-7th order) of surface points as basic features to support retrieval of 3D models. Differently from the case of 2D images, evaluation of moments is not affected by (self-)occlusions.

In order to capture geometric features as well as their arrangement on the object surface, in [8] description and retrieval of 3D objects is accomplished through a combination of warping and projection. However, this method can be applied only to objects whose surface defines the boundary of a simply connected 3D region. Moreover, warping may introduce irregular deformation of the object surface before its projection on a 2D map.

Correlograms have been previously used with success for retrieval of images based on color content [14]. In particular, with respect to description based on histograms of local features, correlograms enable also encoding of information about the relative localization of local features. In [15], histograms of surface curvature have been used to support description and retrieval of 3D objects. However, since histograms do not include any spatial information, the system is liable to false positives.

In this paper, we present a model for representation and retrieval of 3D objects based on curvature correlograms. Correlograms are used to encode information about curva-

ture values and their localization on the object surface. For this peculiarity, description of 3D objects based on correlograms of curvature proves to be very effective for the purpose of content based retrieval of 3D objects.

This paper is organized as follows: in Sect.2 representation of object structure through curvature correlograms is presented; in Sect.3 some distance measures are defined to be used for computing the similarity between two curvature correlograms; finally, in Sect.4 experimental results and performance comparison with alternative approaches are presented.

2 Computation of Curvature Correlograms

High resolution 3D models obtained through scanning of real world objects are often affected by high frequency noise, due to either the scanning device or the subsequent registration process. Hence, smoothing is required to cope with such models for the purpose of extracting their salient features. This is especially true if salient features are related to differential properties of mesh surface (e.g. surface curvature).

Selection of a smoothing filter is a critical step, as application of some filters entails changes in the shape of the models. In the proposed solution, we adopted the filter first proposed by Taubin [10]. This filter, also known as $\lambda|\mu$ filter, operates iteratively, and interleaves a Laplacian smoothing weighed by λ with a second smoothing weighed with a negative factor μ ($\lambda > 0$, $\mu < -\lambda < 0$). This second step is introduced to preserve the model's original shape.

Let $\mathcal{M}$ be a mesh. We denote with E, V e F, the sets of all *edges*, *vertices* and *faces* of the mesh. With N_V, N_E and N_F, we denote the cardinality of sets V, E and F.

Given a vertex $v \in \mathcal{M}$, the principal curvature of $\mathcal{M}$ at vertex v is indicated as $k_1(v)$ and $k_2(v)$. The mean curvature $\bar{k}_v$ is related to the principal curvature $k_1(v)$ and $k_2(v)$ by the equation:

$$\bar{k}_v = \frac{k_1(v) + k_2(v)}{2}$$

Details about computation of the principal and mean curvature for a mesh can be found in [11].

Values of the mean curvature are quantized into $2N + 1$ classes of discrete values. For this purpose, a quantization module processes the mean curvature value through a stair-step function so that many neighboring values are mapped to one output value:

$$\mathcal{Q}(\bar{k}) = \begin{cases} N\Delta & \text{if } \bar{k} \geq N\Delta \\ i\Delta & \text{if } \bar{k} \in [i\Delta, (i+1)\Delta) \\ -i\Delta & \text{if } \bar{k} \in [-(i+1)\Delta, -i\Delta) \\ -N\Delta & \text{if } \bar{k} \leq -N\Delta \end{cases} \tag{1}$$

with $i \in \{0, \ldots, N-1\}$ and Δ a suitable quantization parameter (in the experiments reported in Sect.4 $N = 100$ and $\Delta = 0.15$). Function $\mathcal{Q}(\cdot)$ quantize values of $\bar{k}$ into $2N + 1$ distinct classes $\{c_i\}_{i=-N}^{N}$.

To simplify notation, $v \in \mathcal{M}_i$ is synonymous with $v \in \mathcal{M}$ and $\mathcal{Q}(\bar{k}_v) = c_i$.

Definition 1 (Histogram of Curvature). *Given a quantization of curvature values into* $2N+1$ *classes* $\{c_i\}_{i=-N}^{N}$*, the histogram of curvature* $h_{c_i}(\mathcal{M})$ *of the mesh* $\mathcal{M}$ *is defined as:*

$$h_{c_i}(\mathcal{M}) = N_V \cdot \Pr_{v_i \in \mathcal{M}}[v_i \in \mathcal{M}_i]$$

being N_V the number of mesh vertices.

In doing so, $h_{c_i}(\mathcal{M})/N_V$ is the probability that the quantized curvature of a generic vertex of the mesh belongs to classes c_i.

The correlogram of curvature is defined with respect to a predefined distance value δ. In particular, the curvature correlogram $\gamma_{c_i c_j}^{(\delta)}$ of a mesh $\mathcal{M}$ is defined as:

$$\gamma_{c_i,c_j}^{(\delta)}(\mathcal{M}) = \Pr_{v_1,v_2 \in \mathcal{M}}[(v_1 \in \mathcal{M}_{c_i}, v_2 \in \mathcal{M}_{c_j}) \mid ||v_1 - v_2|| = \delta]$$

In this way, $\gamma_{c_i,c_j}^{(\delta)}(\mathcal{M})$ is the probability that two vertices that are δ far away from each other have curvature belonging to class c_i and c_j, respectively.

Ideally, $||v_1 - v_2||$ should be the geodesic distance between vertices v_1 and v_2. However, this can be approximated with the $k-$ring distance if the mesh $\mathcal{M}$ is regular and triangulated [12].

Definition 2 (1-ring). *Given a generic vertex* $v_i \in \mathcal{M}$*, the* neighborhood *or* 1-ring *of* v_i *is the set:*

$$V^{v_i} = \{v_j \in \mathcal{M} : \exists e_{ij} \in E\}$$

being E *the set of all mesh edges (if* $e_{ij} \in E$ *there is an edge that links vertices* v_i *and* v_j*).*

The set V^{v_i} can be easily computed using the morphological operator *dilate* [13]:

$$V^{v_i} = dilate(v_i)$$

Through the dilate operator, the concept of *1-ring* can be used to define, recursively, generic k^{th} order neighborhood:

$$ring_k = dilate^k \cap dilate^{k-1}$$

Definition of k^{th} order neighborhood enables definition of a true metric between vertices of a mesh. This metric can be used for the purpose of computing curvature correlograms as an approximation of the usual geodesic distance (that is computationally much more demanding). According to this, we define the $k-$ring distance between two mesh vertices as $d_{ring}(v_1, v_2) = k$ if $v_2 \in ring_k(v_1)$.

Function $d_{ring}(v_1, v_2) = k$ is a true metric, in fact:

1. $d_{ring}(u, v) \geq 0$, and $d_{ring}(u, v) = 0$ if and only if $u = v$.
2. $d_{ring}(u, v) = d_{ring}(v, u)$
3. $\forall w \in \mathcal{M}\ \ d(u, v) \leq d(u, w) + d(w, v)$

Based on the $d_{ring}(\cdot)$ distance, the correlogram of curvature can be redefined as follows:

$$\gamma_{c_i,c_j}^{(k)}(\mathcal{M}) = \Pr_{v_1,v_2 \in \mathcal{M}}[(v_1 \in \mathcal{M}_{c_i}, v_2 \in \mathcal{M}_{c_j}) \,|d_{ring}(v_1, v_2) = k]$$

Figs.1(a)-(b) show the correlograms of three models derived from two different model categories, *statue* and *dinosaur*, respectively.

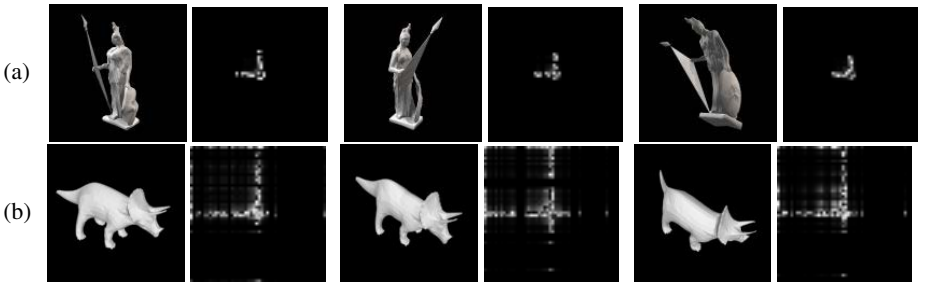


Fig. 1. In (a) and (b) correlograms of three models, taken from two distinct categories, namely statue and dinosaur, are shown.

3 Matching Curvature Correlograms

Several distance measures have been proposed to compute the dissimilarity of distribution functions. In order to compute the similarity between curvature correlograms of two distinct meshes $\gamma^{(k)}_{c_i,c_j}(\mathcal{M}_1)$ and $\gamma^{(k)}_{c_i,c_j}(\mathcal{M}_2)$ we experimented the following distance measures:

Minkowsky-form distance

$$d_{\mathcal{L}_p} = \left[\sum_{i,j=-N}^{N} \left| \gamma^{(k)}_{c_i,c_j}(\mathcal{M}_1) - \gamma^{(k)}_{c_i,c_j}(\mathcal{M}_2) \right|^p \right]^{1/p}$$

Histogram intersection

$$d_{HI} = 1 - \frac{\sum_{i,j=-N}^{N} \min\left(\gamma^{(k)}_{c_i,c_j}(\mathcal{M}_1), \gamma^{(k)}_{c_i,c_j}(\mathcal{M}_2) \right)}{\sum_{i,j=-N}^{N} \gamma^{(k)}_{c_i,c_j}(\mathcal{M}_2)}$$

χ^2-statistics

$$d_{\chi^2} = \sum_{i,j=-N}^{N} \frac{\left(\gamma^{(k)}_{c_i,c_j}(\mathcal{M}_1) - \gamma^{(k)}_{c_i,c_j}(\mathcal{M}_2) \right)^2}{2\left(\gamma^{(k)}_{c_i,c_j}(\mathcal{M}_1) + \gamma^{(k)}_{c_i,c_j}(\mathcal{M}_2) \right)}$$

Kullback-Leibler divergence

$$d_{KL} = \sum_{i,j=-N}^{N} \gamma^{(k)}_{c_i,c_j}(\mathcal{M}_1) log \frac{\gamma^{(k)}_{c_i,c_j}(\mathcal{M}_1)}{\gamma^{(k)}_{c_i,c_j}(\mathcal{M}_2)}$$

Using a groundtruth database, the above distance measures have been compared in terms of precision and recall figures. Results of this analysis (not reported in this paper for lack of space) suggest that the best performance is achieved by using χ^2-statistics to measure the distance between curvature correlograms.

4 Experimental Results

Approximately 300 models were collected to build the test database. These comprise four classes of models: taken from the web, manually authored (with a 3D CAD soft-

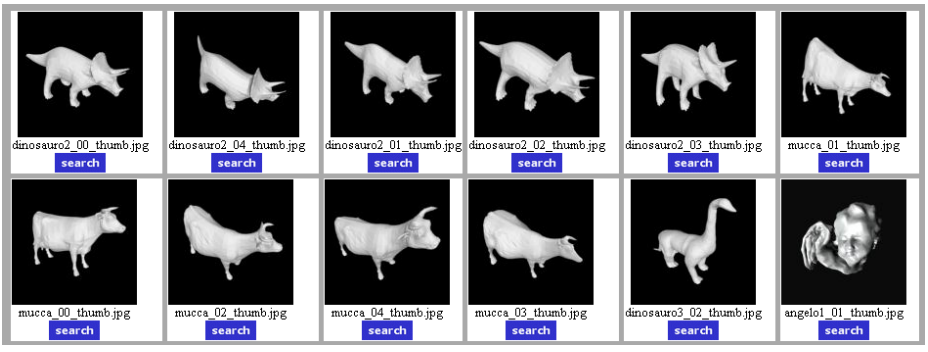


Fig. 2. A retrieval example, using the model of a dinosaur as the query. Other models of dinosaurs were retrieved first, followed by models of other objects which display similar features.

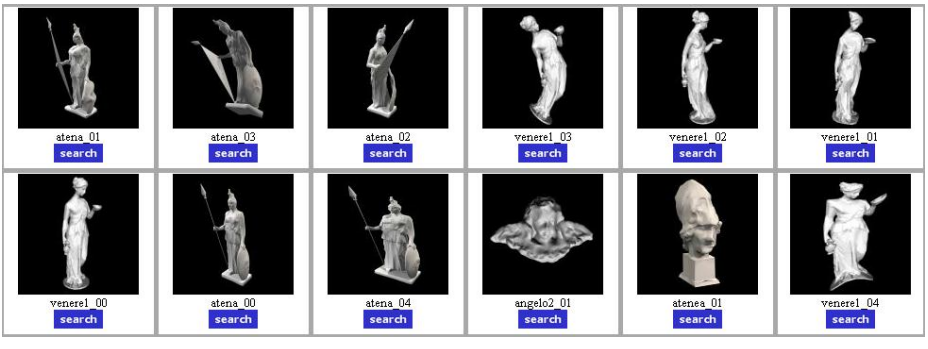


Fig. 3. The model of a statue is used as query. Other models of the same statue are retrieved, as well as models which display busts or different statues.

ware), high quality 3D scans from the De Espona 3D Models Encyclopedia[1], and variations of the previous three classes (obtained through deformation or application of noise, which caused points surface to be moved from their original locations). Content descriptors, in the form of curvature correlograms, were then extracted from database objects and added to the index.

Fig.2 shows a retrieval example where the model of a dinosaur is used as a query. The result set displays all models of similar animals in the first positions. These animals all share some major features such as four legs, a tail and two horns.
Retrieval results for a query representing a statue are shown in Fig.3. In this case, models of statues and busts are ranked in the first positions of the retrieval set.

In Fig. 4, the precision-recall curve of the proposed approach is compared with the curve of alternative approaches. In particular, approaches based on 3D geometric moments [7], curvature histograms [15], curvature maps [8] and spin images [9] are considered. The proposed solution shows an improved performance with respect to all the four alternative approaches.

[1] `http://www.deespona.com`

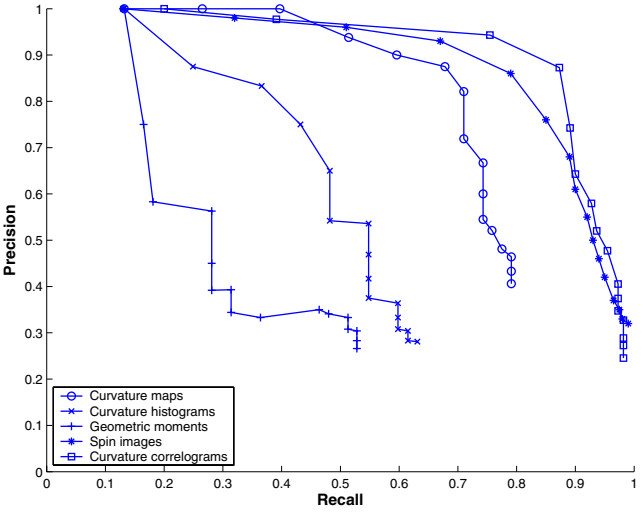


Fig. 4. The average Precision-Recall curves over a given set of queries show an improved performance of the correlograms with respect to four alternative approaches.

5 Conclusions and Future Work

In this paper we have presented an approach to retrieval by content of 3D objects based on curvature correlograms. The main advantage of correlograms relates to their ability to encode not only distribution of features but also their arrangement on the object surface. Experimental results have shown that the proposed solution is well suited for the purpose of retrieval. Furthermore, results showed that the approach performs better than previous approaches to content-based retrieval of 3D objects. Future work will address extension of correlograms to deal with multiresolution descriptors as well as the definition of suitable distance measures to cope with retrieval by object parts.

References

1. S. Mahmoudi, M. Daoudi. “3D models retrieval by using characteristic views,” in *Proc. of 16th Int’l Conf. on Pattern Recognition*, Vol.2, pp.457-460, 11-15 Aug, 2002.
2. R. Ohbuchi, M. Nakazawa, T. Takei. “Retrieving 3D Shapes based on Their Appearance,” in *Proc. of MIR’03*, Berkeley, CA, USA, Nov. 2003, pp.39-46.
3. H.P. Kriegel, T. Seidl. “Approximation-Based Similarity Search for 3D Surface Segments,” *GeoInformatica Journal*, 2(2):113-147, Kluwer Academic Publisher, 1998.
4. E. Paquet, M. Rioux. “Nefertiti: a query by content system for three-dimensional model and image database management,” *Image Vision and Computing*, 17(2):157-166, 1999.
5. I. Kolonias, D. Tzovaras, S. Malassiotis, M. G. Strintzis, “Content-Based Similarity Search of VRML Models Using Shape Descriptors”, in *Proc. of International Workshop on Content-Based Multimedia Indexing*, Brescia (I), September 19-21, 2001.

6. F. Mokhtarian, N. Khalili, P. Yeun. “Multi-scale free-form 3D object recognition using 3D models,” *Image and Vision Computing*, 19(5):271-281, 2001.
7. M. Elad, A. Tal, S. Ar. “Content Based Retrieval of VRML Objects - An Iterative and Interactive Approach,” *EG Multimedia*, September 2001, 97-108.
8. J. Assfalg, A. Del Bimbo, P. Pala. “Curvature Maps for 3D CBR”, in *Proc. of Int’l Conf. on Multimedia and Expo (ICME’03)*, Baltimore (MD), July 2003.
9. J. Assfalg, G. D’Amico, A. Del Bimbo, P. Pala. “3D content-based retrieval with spin images,” in *Proc. of Int’l Conf. on Multimedia and Expo (ICME’04)*, Taipei, Taiwan, June 27-30, 2004.
10. G. Taubin. “A Signal Processing Approach to Fair Surface Design,” *Computer Graphics (Annual Conference Series)*, 29:351–358, 1995.
11. G. Taubin. “Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation,” in *Proc. of Fifth International Conference on Computer Vision (ICCV’95)*, pp.902-907.
12. Mathieu Desbrun, Mark Meyer, Peter Schroder and Alan H. Barr. “Discrete Differential-Geometry Operators in nD”, Caltech, 2000.
13. Christian Rössl, Leif Kobbelt, Hans-Peter Seidel. “Extraction of Feature Lines on Triangulated Surfaces using Morphological Operators,” in *Smart Graphics*, Proceedings of the 2000 AAAI Symposium
14. J. Huang, R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih. “Statial Color Indexing and Application,” in *Internation Journal of Computer Vision*, Vol. 35, pp. 245-268, 1999
15. G. Hetzel, B. Leibe, P. Levi, B. Schiele. “3D Object Recognition from Range Images using Local Feature Histograms,” in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition (CVPR’01)*, Kauai Marriott, Hawaii, Dec. 9-14, 2001.

3D Surface Reconstruction of a Moving Object in the Presence of Specular Reflection

Atsuto Maki

Graduate School of Informatics, Kyoto University,
Kyoto, 606-8501, Japan

Abstract. We present a new scheme for 3D surface reconstruction of a moving object in the presence of specular reflection. We basically search for the depth at each point on the surface of the object while exploiting the recently proposed geotensity constraint [7] that accurately governs the relationship between four or more images of a moving object in spite of the illumination variance due to object motion. The thrust of this paper is then to extend the availability of the geotensity constraint to the case that specularities are also present. The key idea is to utilise the fact that highlights shift on the surface due to object motion. I.e., we employ five or more images as inputs, and interchangeably utilise a certain intensity subset consisting of four projected intensities which is the least influenced by the specular component. We illustrate the relevancy of our simple algorithm also through experiments.

1 Introduction

Given a set of images, in each of which an object is viewed from a different direction, the fundamental issue in extracting 3D information of the object out of 2D images is to match corresponding points in those images so that these points are the projections of an identical point on the surface of it. For the point correspondence, typically exploited is the constraint that the corresponding parts of the images have equivalent image intensities, regarding the variation in illumination as noise. It has been successfully applied to stereo (see for example [4,5]) where two images are taken simultaneously as the lighting of the object is identical in each image. However, when we consider replacing the stereo camera with a single camera observing an object in motion, unfortunately the constraint is nearly always invalid as non-uniform lighting causes the intensity at a specific location on the surface of an object to change as the object moves. Among the few efforts for this issue, whereas *photometric motion* [9] treated the illumination variance due to object motion in terms of optical flow, *geotensity* constraint[1] [7] has been recently derived to overcome the problem with respect to camera geometry, and to replace the constant intensity constraint. Based on the notion of linear intensity subspaces [10], the geotensity constraint governs the relationship between four (or more) images of a moving object, and it can be computed and applied automatically to the task of 3D surface reconstruction.

[1] *Geotensity* stands for "geometrically corresponding pixel intensity."

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 867–875, 2005.

The algorithm for surface reconstruction using geotensity constraint proceeds basically in two stages. The first stage is to derive the parameters of the geotensity constraint by analysing coordinates and image intensities of some sample points on the object in motion. That is, computing structure from motion obtains the geometric parameters of the situation, whereas computing the linear image subspace obtains the lighting parameters of the situation. By combining both sets of parameters we arrive at the geotensity constraint. Using the same set of images, the second stage is to take each pixel in an arbitrary reference image in turn and search for the depth along the ray from the optical centre of the camera passing through the pixel. The depth is evaluated by measuring the agreement of the entire set of projected intensities of a point on the object surface with the geotensity constraint.

Although the availability of the constraint was limited in principle to Lambertian surface as is the case with the other sophisticated approaches [11,13], the thrust of this paper is to extend it to the situation that the object surface partly takes on specular reflection. In the case of stereo, in the presence of specular reflection, it has been shown possible to determine trinocular configurations such that at least one stereo pair can provide correct depth estimate at each scene point visible to all cameras [3]. In contrast, given a single static camera, we propose to employ five or more images as inputs, and interchangeably utilise a certain *intensity subset* consisting of four projected image intensities that is the least influenced by the specular component of surface reflection. The strategy is motivated by the fact that the specularities shift on the surface thanks to the object motion. Through experiments we show that the proposed scheme indeed improves the depth estimate compared to the results by other standard algorithms in terms of dealing with specular reflection.

2 Preliminaries

As the background we first consider some issues respecting geometry and image intensity that form the basis of the geotensity constraint. What we initially need is to find some number of corresponding sample points by an independent mechanism as seen for example in [1,14]. Given point correspondence for some sample points, we can derive a constraint on geometry by the coordinates, and also a photometric constraint by observing the intensities on these points.

Solving for geometry. In this paper, for simplicity, we will concern ourselves with the *affine* and *scaled-orthographic* camera models for projection. Consider the i^{th} world point $\mathbf{X}_i = (X_i, Y_i, Z_i)^\top$ on the surface of an object projected to image point $\mathbf{x}_i(j) = (x_i(j), y_i(j))^\top$ in the j^{th} frame. The affine camera model defines this projection to be $\mathbf{x}_i(j) = \mathbf{M}(j)\mathbf{X}_i + \mathbf{t}(j)$, where $\mathbf{M}(j)$, an arbitrary 2×3 matrix, and $\mathbf{t}(j)$, an arbitrary 2 vector, encode the motion parameters of the object. The solution to the structure from motion problem using singular value decomposition is well known for this case; given at least four point trajectories $\mathbf{x}_i(j)$ observed through at least two frames the set $\mathbf{M}(j)$, $\mathbf{X}_i$ and $\mathbf{t}(j)$ can be uniquely recovered up to an arbitrary affine ambiguity [12]. The result is affine structure.

Given the solution to structure from motion using the affine camera model, the Euclidean structure and motion parameters fitting the weak perspective camera model can be recovered. A result of choosing the first frame to be canonical is that the structure vectors have the form, $\mathbf{X}_i = (\mathbf{x}_i^\top(1), Z)^\top$, and we can derive the relationship which effectively describes the *epipolar constraint* between two images:

$$\mathbf{x}_i(j) = \mathbf{M}(j) \begin{pmatrix} \mathbf{x}_i(1) \\ Z \end{pmatrix} + \mathbf{t}(j) \ . \tag{1}$$

Solving for image intensity. Assuming a static camera and a single point light source, we consider the intensity $I_i(j)$ of the i^{th} point on the surface of a moving object projected into the j^{th} image. For Lambertian surface, since the intensity can be equivalently represented as if it were generated on the surface of a static object under inversely moving point light source, we can then express $I_i(j)$ in terms of the image formation equation process so that

$$I_i(j) = \max(\mathbf{b}_i{}^\top \mathbf{s}(j), 0) \ . \tag{2}$$

The 3-vector $\mathbf{b}_i$ is defined at the first frame to be the product of the albedo with the inward facing unit normal for the i^{th} point whereas the 3-vector $\mathbf{s}(j)$ is the product of the strength of the light source with the unit vector for its direction. Note that $\mathbf{s}(j) = \mathbf{R}^\top(j)\ \mathbf{s}(1)$, where the 3×3 matrix, $\mathbf{R}(j)$, is the rotation of the object from the first frame, which is taken to be the reference, to the j^{th} frame. Multiplication of $\mathbf{R}^\top(j)$ represents virtually inverse rotation of the light source. The rotation matrix is directly computed from the 2×3 matrix $\mathbf{M}(j)$ that is given above by solving for the structure from motion problem. Although the maximum operator zeroes negative components [6] which correspond to the shadowed surface points, we assume that there are no shadows in the subsequent formulations.

Given the correspondence for n feature points through m images, we record the corresponding pixel intensities, $I_i(j)$, in an $n \times m$ matrix as $\mathbf{I} = (I_i(j))$, which we call the illumination matrix. Then, we can form the matrix equation, $\mathbf{I} = \mathbf{BS}$, where $\mathbf{B}$ is an $n \times 3$ matrix containing the rows $\mathbf{b}_i^\top$, and $\mathbf{S}$ is a $3 \times m$ matrix containing the columns $\mathbf{s}(j)$. Then, the familiar form for solution by singular value decomposition to obtain a rank 3 approximation to the matrix $\mathbf{I}$ is such that

$$\mathbf{I} = \breve{\mathbf{B}}\breve{\mathbf{S}} = (\breve{\mathbf{B}}\mathbf{A}^{-1})(\mathbf{A}\breve{\mathbf{S}}) \ . \tag{3}$$

As is well known, the solution is unique up to an arbitrary invertible 3×3 transformation $\mathbf{A}$ which transforms $\breve{\mathbf{S}}$ into $\mathbf{S}$, or $\breve{\mathbf{s}}(j)$ into $\mathbf{s}(j)$, for each column by

$$\mathbf{S} = \mathbf{A}\breve{\mathbf{S}} \ , \quad \mathbf{s}(j) = \mathbf{A}\breve{\mathbf{s}}(j) \ , \tag{4}$$

where $\breve{\mathbf{s}}(j)$ denotes each column of $\breve{\mathbf{S}}$.

3 The Geotensity Constraint for Correspondence Search

Although a thorough description of the geotensity constraint can be found in [7], in this section we briefly review the constraint while reformulating it so as to

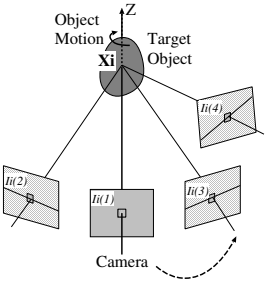


Fig. 1. Geotensity constraint. The intensity of world point $\mathbf{X}_i$ projected into the first image, $I_i(1)$, is represented by a unique linear combination of the intensities of the same point projected in the other three images, $I_i(2) \cdots I_i(4)$ for all points i.

prepare for the extension for the case with specularities which we will introduce in Section 4. The term geotensity constraint accounts for a constraint between four or more images of an object from different views under static lighting conditions. This concept is schematically depicted in Figure 1 by replacing object motion with an imaginary, and coherent motion of the camera and the light source.

The conditions for applying the geotensity constraint to surface reconstruction are as follows: (i) The scene consists of a single moving object that is convex (therefore no self-shadowing). (ii) There is a single distant light source. (iii) The object has Lambertian surface properties while the surface may or may not be textured. However, the condition (iii) will be relaxed in Section 4.

Evaluating the set of intensities for correspondence. At each pixel, $\mathbf{x}$, in the first image, to search for the depth Z we can recall equation 1 for the geometric constraint imposed on a sequence of images so that

$$I(j; \mathbf{x}, Z) = I[\mathbf{x}(j)] = I[\mathbf{M}(j) \begin{pmatrix} \mathbf{x}(1) \\ Z \end{pmatrix} + \mathbf{t}(j)] \ . \tag{5}$$

$I(j; \mathbf{x}, Z)(j = 1, ..., m; m \geq 4)$ indicates the set of image intensities in the j^{th} frame at the coordinates determined by $\mathbf{x}$ in the first image, guess of depth Z, and the motion parameters $\mathbf{M}(j)$ and $\mathbf{t}(j)$. The task is now to evaluate the set of intensities $I(j; \mathbf{x}, Z)$. If full Euclidean lighting conditions have been recovered in advance so that $\mathbf{s}(j)$ is known without ambiguity, we define $\hat{\mathbf{b}}^\top$ as

$$\hat{\mathbf{b}}^\top = [I(1), ..., I(m)] \, \mathbf{S}^\top (\mathbf{S}\mathbf{S}^\top)^{-1} \tag{6}$$

where $\mathbf{S}$ is a $3 \times m$ matrix containing the columns $\mathbf{s}(j)(j = 1, ..., m)$. For a single light source with all images taken with the light source in the so-called *bright cell* [2], the estimated values of the intensities would then be

$$\hat{I}(j; \mathbf{x}, Z) = \hat{\mathbf{b}}^\top \mathbf{s}(j) \ . \tag{7}$$

[2] The cell of light source directions that illuminate all points on the object [2].

It should be noted that exactly the same estimation of $\hat{I}(j;\mathbf{x},Z)$ is available also in the case that the light source direction is determined only up to the ambiguity. This is easily confirmed by substituting equation 4 to equation 6, and then to equation 7, where matrix $\mathbf{A}$ turns out to be cancelled. Thus, the set of equations 6 and 7 can be represented equivalently as

$$\hat{I}(j;\mathbf{x},Z) = \hat{\mathbf{b}}^{\top}\check{\mathbf{s}}(j) \ , \quad \hat{\mathbf{b}}^{\top} = [I(1),...,I(m)]\,\check{\mathbf{S}}^{\top}(\check{\mathbf{S}}\check{\mathbf{S}}^{\top})^{-1} \ . \tag{8}$$

Estimating $\hat{I}(j;\mathbf{x},Z)$ by equation 8, we can define the error function to evaluate the set of intensities $I(j;\mathbf{x},Z)$ as

$$E(\mathbf{x},Z) = \sum_{j=1}^{m}(I(j;\mathbf{x},Z) - \hat{I}(j;\mathbf{x},Z))^2 \ . \tag{9}$$

Computing the depth. At each pixel, $\mathbf{x}$, in the first reference image we measure the error, E, in the geotensity constraint at regular small intervals of depth, Z. When the depth is correct we expect the error to approach zero and when it is incorrect we expect the error to be large. The geotensity constraint can be therefore stated simply as $E(\mathbf{x},Z) = 0$. It is clear that as the depth parameter is varied the location of the corresponding points in each image will trace out the corresponding epipolar line in each image. We then choose such depth Z that minimises the error $E(\mathbf{x},Z)$ as the depth estimate.

4 Dealing with Specular Reflection

The technique of evaluating the set of projected intensities discussed in the previous section is deficient in dealing with specular reflection. This is because the notion of linear image basis which plays the essential role is only valid for Lambertian surface. In order to cope with specular reflection within the same framework, we propose to employ five or more images as inputs, and interchangeably utilise a certain subset consisting of four intensities, which are required at the minimum for the geotensity constraint to be applied.

Since we consider a moving object as the target, each surface point may or may not have specularity depending on the object's pose as a specular lobe in general has some certain limited directional range. We may thus expect that a certain subset of projected intensities exists that satisfies the geotensity constraint to a reasonable extent. At least it is very unlikely that some specular reflection is added to an identical surface point throughout all the input image frames (except the case the object consists of a particular kind of material such as metal). The problem to cope with the situation is then attributed to that of finding an *intensity subset* which is the least influenced by specular component of surface reflection.

Algorithm description. As the simplest case, let us consider that five images are given as input so that $m = 5$, and assume that specular reflection occurs mostly in either of them due to object motion. Given an $n \times m$ illumination

matrix as $\mathbf{I} = (I_i(j))$, we define m different $n \times (m-1)$ *illumination submatrices*, $\mathbf{I}_{\bar{k}}(k = 1, ..., m)$, by skipping the k^{th} column, $\mathbf{I}(k)$, such as

$$\mathbf{I}_{\bar{k}} = [\mathbf{I}(1) \ ... \ \mathbf{I}(k-1) \ \mathbf{I}(k+1) \ ... \ \mathbf{I}(m)] \ . \tag{10}$$

We then recall equation 3 and compute a rank 3 approximation to each $\mathbf{I}_{\bar{k}}$,

$$\mathbf{I}_{\bar{k}} = \breve{\mathbf{B}}_{\bar{k}} \breve{\mathbf{S}}_{\bar{k}} = (\breve{\mathbf{B}}_{\bar{k}} \mathbf{A}^{-1})(\mathbf{A} \breve{\mathbf{S}}_{\bar{k}}) \ , \tag{11}$$

where $\breve{\mathbf{S}}_{\bar{k}}(k = 1, ..., m)$ in this case is a $3 \times (m-1)$ matrix containing the columns $\breve{\mathbf{s}}(j)(j = 1, ..., m-1)$. Importantly, for computing each $\breve{\mathbf{S}}_{\bar{k}}$ we employ RANSAC, a robust random sampling and consensus technique, to ensure that artifacts which are caused by specularities (or some intensities not fulfilling the assumed conditions, e.g. self-shadowing) do not distort the correct solution.

At the correct depth in correspondence search, an intensity subset, $I(j; \mathbf{x}, Z)(j = 1, ..., m-1)$, excluding the k^{th} element should be properly validated with corresponding light source matrix $\breve{\mathbf{S}}_{\bar{k}}$ if specular reflection occurs in the k^{th} frame. Hence, with each $\breve{\mathbf{S}}_{\bar{k}}$, we can exploit equation 8 to estimate $\hat{I}(j; \mathbf{x}, Z)$ $(j = 1, ..., m-1)$ and then evaluate the subset of intensities by

$$E_{\bar{k}}(\mathbf{x}, Z) = \sum_{j=1, j \neq k}^{m} (I(j; \mathbf{x}, Z) - \hat{I}(j; \mathbf{x}, Z))^2 \ . \tag{12}$$

Namely, we obtain m different $E_{\bar{k}}(\mathbf{x}, Z)(k = 1, ..., m)$ due to m different candidates of intensity subset (together with corresponding $\breve{\mathbf{S}}_{\bar{k}}$) and one of them must be appropriate according to the above assumption.

In order to search for the depth Z, we need to judge the correct error among those defined by equation 12. Although it is not possible to know which subset of projected intensities is the least influenced by specularity in advance, since the error $E_{\bar{k}}(\mathbf{x}, Z)$ generally becomes larger with specularity, it is sensible to choose the smallest one so that

$$E(\mathbf{x}, Z) = \min_{k} E_{\bar{k}}(\mathbf{x}, Z) \ . \tag{13}$$

Just as in the case without specular reflection, we expect the error to approach zero when the depth is correct, and choose such depth Z that minimises the error $E(\mathbf{x}, Z)$ as the depth estimate. Our algorithm for estimating the depth in the presence of specularities can be summarised as following:

1° Decompose each illumination submatrix $\mathbf{I}_{\bar{k}}$ using SVD and yield $\breve{\mathbf{S}}_{\bar{k}}$.
2° At point $\mathbf{x}$, measure $I(j; \mathbf{x}, Z)$ by equation 5 for a guess of depth Z.
3° Estimate $\hat{I}(j; \mathbf{x}, Z)$ by equation 8, for all $\breve{\mathbf{S}}_{\bar{k}}$ using $I(j; \mathbf{x}, Z)(j \neq k)$.
4° Compute $E_{\bar{k}}(\mathbf{x}, Z)$ and then $E(\mathbf{x}, Z)$ by equations 12 and 13.
5° Choose such depth Z that minimises $E(\mathbf{x}, Z)$ as the depth estimate.

Alternatively, we may simplify step 3° by estimating $\hat{I}(j; \mathbf{x}, Z)$ just in one way, rather than examining it for all the possible candidates of $\breve{\mathbf{S}}_{\bar{k}}$. That is, we simply

Fig. 2. Example input images of a Dharma doll. Captured in different poses under a point light source placed beside the camera (Nikon D70). Image size: 320 × 400 pixels. It is observed that the specularities shift on the surface thanks to the object motion.

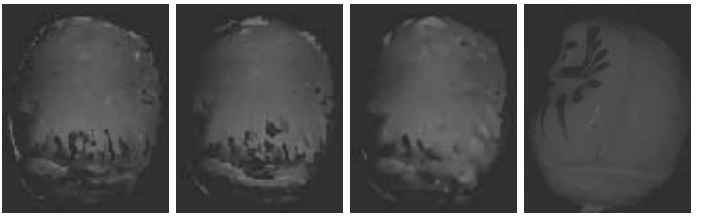


Fig. 3. Three depth maps estimated referring the midmost image in Figure 2 (the lighter, the closer). From the left: differencing, the geotensity, and the geotensity with *intensity subset*. The profile of the Dharma doll is also shown for reference.

determine k such that the k^{th} element has the highest value among the set of projected intensities, and compute $E_{\bar{k}}(\mathbf{x}, Z)$ accordingly. It is based on the observation that setting the element of highest intensity aside must be the choice for the solution to be the least influenced by specular reflection at the correct depth. As we can then skip estimating different $E_{\bar{k}}(\mathbf{x}, Z)$ in step 4°, it is also less expensive in the computational viewpoint.

Finally, the basic principle in the proposition applies to a larger number of input images, $m > 5$, and we then need choose the size of the intensity subset. Although a large size implies higher tolerance against noise as long as specular reflection appears just in one frame as assumed above, a crude observation tells that it rather is optimal to keep the size minimum, i.e. four elements, to allow a higher probability of including no specularities. However, the trade-off analysis is also motion dependent and it is beyond the scope of this paper.

5 Experiments

We illustrate the performance of the proposed scheme using an object whose surface is moderately glossy (see Figure 2). The input images were taken while the object pose varies under a point light source. As can be observed, specularities appear on different part of the surface according to the object pose relative to the light source. Figure 3 shows the resulting depth map computed for the midmost input image in Figure 2. For comparison, we carried out the depth search

by three different algorithms within the common framework, i.e., differencing, the geotensity constraint, and the proposed – the geotensity constraint using the intensity subset. In each case we used a 15×15 template for the search to suppress the error arising from image noise. It turned out that the proposed method allows an improved estimation whereas the other two tend to fail more severely in the region where the surface is either with little texture, or with specularity such as on the top of the *belly*, although the erroneous estimates in the bottom (conceivably due to complicated inter-reflection) are of an issue for all of them.

6 Conclusion

For the problem of 3D object surface reconstruction by a single static camera, we have proposed to relax the condition for geotensity constraint to be applicable to the case that specular reflection is added to the object surface. The key idea is to utilise such a certain subset of intensities that is the least influenced by specular reflection. To our knowledge it is one of the few trials of developing correspondence scheme in the presence of specularity as well as under varying image intensity due to object motion. Although we have illustrated our scheme only for the case of using five input images, the principle also applies to the case that a larger number of images are available. However, we further need some investigations as to how the size of the intensity subset in such cases can be optimally determined, which is one of the issues for future work. Another interesting application will be to generate illumination image basis [8] in the presence of specular reflection on the basis of the proposed scheme for correspondence.

Acknowledgments. This work is supported in part by MEXT, Japan, under a Grant-in-Aid for Scientific Research (No.16680010) and the research project of MEXT, headed by Professor T. Matsuyama, "Developement of high fidelity digitization software for large-scale and intangible cultural assets".

References

1. P.A. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *4rd ECCV*, pages 683–695, Cambridge, UK, 1996.
2. P.N. Belhumeur and D.J. Kriegman. What is the set of images of an object under all possible illumination conditions? *IJCV*, 28:3:245–260, 1998.
3. D. N. Bhat and S. Nayar. Stereo in the presence of specular reflection. In *5th ICCV*, pages 1086–1092, 1995.
4. F. Devernay and O. Faugeras. Computing dierential properties of 3D shapes from stereoscopic images without 3d models. In *CVPR*, pages 208–213, 1994.
5. P. Fua. Object-centered surface reconstruction: combining multi-image stereo and shading. *IJCV*, 16:35–56, 1995.
6. B.K.P. Horn. *Robot Vision*. The MIT Press, 1992.
7. A. Maki, M. Watanabe, and C.S. Wiles. Geotensity: Combining motion and lighting for 3d surface reconstruction. *IJCV*, 48:2:75–90, 2002.
8. A. Nakashima, A. Maki, and K. Fukui. Constructing illumination image basis from object motion. In *7th ECCV*, pages III:195–209, 2002.

9. A. Pentland. Photometric motion. *IEEE-PAMI*, 13:9:879–890, 1991.
10. A. Shashua. *Geometry and photometry in 3D visual recognition*. PhD thesis, Dept. Brain and Cognitive Science, MIT, 1992.
11. D. Simakov, D. Srolova, and R. Basri. Dense shape reconstruction of a moving object under arbitrary, unknown lighting. In *9th ICCV*, pages 1202–1209, 2003.
12. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9:2:137–154, 1992.
13. M. Weber, A. Blake, and R. Cipolla. Towards a complete dense geometric and photometric reconstruction under varying pose and illumination. In *BMVC*, pages 83–92, 2002.
14. C.S. Wiles, A. Maki, and N. Matsuda. Hyper-patches for 3d model acquisition and tracking. *IEEE-PAMI*, 23:12:1391–1403, 2001.

Fitting 3D Cartesian Models to Faces Using Irradiance and Integrability Constraints

Mario Castelán* and Edwin R. Hancock

Dept. of Computer Science, University of York,
York YO10 5DD, UK
{mario, erh}@cs.york.ac.uk

Abstract. This paper makes two contributions. First, we present an experimental analysis of three different ways of constructing three-dimensional statistical models of faces using Cartesian coordinates, namely, height, surface gradient and one based on Fourier domain basis functions. Second, we test the ability of each of the models for dealing with information provided by shape-from-shading by introducing a simple non-exhaustive parameter adjustment procedure subject to integrability and irradiance constraints. Experiments show that the surface gradient based representation is more robust to noise than the alternative Cartesian representations.

1 Introduction

In their work on eigenfaces, Turk and Pentland were among the the first to explore the use of principal components analysis for performing face recognition [6]. This method can be rendered efficient using the technique described by Sirovich et al.[5] which shows how to avoid explicit covariance matrix computation for large sets of two-dimensional images of objects of the same class. The method involves cropping n examples of images of the same class (i.e. faces) and converting them into vectors $\mathbf{x}$ of size $m \times 1$. The mean example, $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$, is subtracted from each of the vectors $\mathbf{x}$ to form the matrix $\mathbf{X}_{m\times n}$. A set of *temporal* eigenvectors $\tilde{\mathbf{\Psi}}_{n\times n}$ is calculated from the symmetric and non-negative outer product matrix $\mathbf{L}_{n\times n} = \mathbf{X}\mathbf{X}^T$. These eigenvectors are ordered according to the increasing magnitude of their corresponding eigenvalues. Subsequently, the *spatial* eigenvectors, or eigenmodes, $\mathbf{\Psi}_{m\times n} = \mathbf{X}\tilde{\mathbf{\Psi}}$ are generated. Every example can be recovered as a linear combination of a set of shape coefficients α with the eigenmodes $\mathbf{\Psi}$, as $E = \bar{\mathbf{x}} + \sum_{i=1}^{l} \alpha_i \mathbf{\Psi}_\mathbf{i}$ where $1 \leq l \leq n$ is the chosen number of modes. By varying the shape coefficients α within ± 3 standard deviations, the resulting image is guaranteed to be a valid example of the class represented by the eigenmodes.

In 2D a set of aligned intensity images of faces is used to construct the eigenmodes, and the image data is usually encoded as a Cartesian long-vector by concatenating the rows or columns of the image. However, if a 3D model is to

* Supported by National Council of Science and Technology (CONACYT), Mexico, under grant No. 141485.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 876–883, 2005.

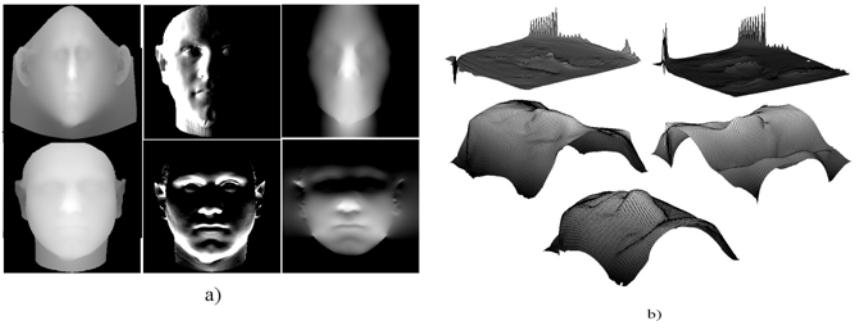


Fig. 1. In the left-hand part of the Figure, a Cylindrical coordinate depth map (top left corner), height (bottom left corner), surface gradient (middle column) and Fourier domain (right-most column) Cartesian representations are shown as intensity plots. The right-hand part of the Figure shows, as height map plots, surface gradient (top row), Fourier basis (middle row) and height (bottom row) representations.

be constructed from range data then there exist alternative ways for representing the data. One of the most commonly used approaches is to adopt a representation that uses cylindrical coordinates. Using cylindrical coordinates, the surface of a human face (or head) can be parameterized by the function $r(\theta, \ell)$, where r is the radius and θ and ℓ are the height and angular coordinates respectively. This representation has been adopted since it captures the linear relations between basis heads, but can lead to ambiguity since different parameter sets can give rise to the same head-shape.

An alternative is to use a Cartesian representation, in which each surface point p is specified by its (x, y, z) co-ordinates, where the z-axis is in the direction of the viewer. The Cartesian coordinates are related to the cylindrical coordinates through $(x, y, z) = (x_0 + r(\theta, \ell)sin\theta, y_0 + \ell, z_0 + r(\theta, \ell)cos\theta)$, for some shift (x_0, y_0, z_0) relating the position of the origin in the two coordinate systems. Cartesian coordinates are not normally used since they introduce high z-variance (height variability) in areas with high spatial variance (i.e. face features such as nose, mouth and eyes). A face depth map in cylindrical coordinates could be thought as an unwrapped version of a depth map expressed in Cartesian coordinates. In Figure 1(a), a cylindrical coordinate depth map (top left corner) is shown together with its corresponding Cartesian coordinate depth map (bottom left corner).

Attick et al.[1] were among the first to build 3D statistical shape models of faces. They developed a shape-from-shading method which recovered shape-coefficients that satisfy the image irradiance equation[1]. The eigenmode surfaces,

[1] The image irradiance equation[4] states that the measured brightness of the image is proportional to the radiance at the corresponding point on the surface. If $n(x, y)$ is the normal vector to the surface of an object at the position (x, y) and s is the light source direction vector, then the perceived brightness can be defined as $I = n \cdot s$. This is the simplest reflectance model and does not take into account albedo and inter-reflection effects.

which they called eigenheads, were parameterized in cylindrical coordinates. Later, Vetter et al.[7] decoupled texture from shape, and performed PCA on the two components separately. Using facial feature correspondences, they developed a model that could be fitted to image data. Although these methods sacrifice efficiency and simplicity since parameter tuning is performed using exhaustive search, they deliver accurate photo-realistic results. Recently, Dogvard and Basri[2] constructed a Cartesian model by combining statistical methods and symmetry constraints. Here the aim was to express the surface gradient in terms of the set of deformation coefficients. This allows shape-from-shading to be transformed into a linear system of equations, that can be simply solved for the shape coefficients and used to reconstruct the height function for the face, subject to symmetry constraints. Although accuracy is sacrificed, the method is computationally efficient.

Encouraged by Dogvard and Basri's work, in this paper our aim is to explore the use of alternative Cartesian representations for constructing 3D statistical models of faces. We also present an intuitive and straightforward method for shape coefficient adjustment which draws on the geometric shape-from-shading method of Worthington and Hancock [8]. In this way we avoid the use of costly heuristic search methods.

The paper is organized as follows: In Section 2 we give a brief explanation about the different Cartesian representations used. Section 3 describes how each of the models was constructed. The experimental results obtained using a data-driven shape coefficients search approach are presented in section 4. Finally, we present conclusions and suggest some possible lines for future work.

2 The Cartesian Representations

We have explored the use of three alternative Cartesian coordinate representations to construct the statistical models. These are height, surface gradient and a combination of these two, which we refer to as the Fourier domain method.

The *Height* representation is based on the surface height function $Z(x, y)$. An example is shown in Figure 1(a),leftmost column, bottom row, where intensity is used to represent the height map of a face.

The *Surface Gradient* representation is based on the directional partial derivatives of the height function $p = \frac{\partial Z(x,y)}{\partial x}$ and $q = \frac{\partial Z(x,y)}{\partial y}$. The set of first partial derivatives of a surface is also known as the gradient space, and it is related to the surface normal through $n = (p, q, -1)$. An example is shown in the middle column of Figure 1(a): x-partial (top) and y-partial (bottom) where intensity is used to represent the surface gradients. From the height function $Z(x, y)$, the first partial derivatives can be calculated through central, forward or backward finite differences.

The *Fourier Basis* representation draws on the concept of Fourier domain integrability for surface height recovery from surface gradient introduced by Frankot and Chellappa [3]. An integrable surface Z (i.e. $\frac{\partial^2 Z(x,y)}{\partial x \partial y} = \frac{\partial^2 Z(x,y)}{\partial y \partial x}$) , can be defined by the basis expansion

$$\tilde{Z}(x,y) = \sum_{\omega \in \Omega} \tilde{C}(\omega)\phi(x,y,\omega) \tag{1}$$

where ω is a two dimensional index belonging to the domain Ω, and $\phi(x,y,\omega)$ is a set of basis functions which are not necessarily mutually orthogonal (i.e. the Fourier transform). The first partial derivatives of $\tilde{Z}$ can also be expressed in terms of this set of basis functions and will share the same set of coefficients $\tilde{C}_x(\omega) = \tilde{C}_y(\omega)$. If a set of first partial derivatives is not integrable then their corresponding transform coefficients will differ from each other (i.e $\hat{C}_x(\omega) \neq \hat{C}_y(\omega)$). The distance between the non-integrable and the integrable partial derivatives can be minimized in the transform domain by making $\hat{C}_x(\omega) = \hat{C}_y(\omega) = \tilde{C}(\omega)$. It is been shown that $\tilde{C}(\omega) = \tilde{C}_1(\omega) + \tilde{C}_2(\omega)$, with

$$\tilde{C}_1(\omega) = \frac{P_x(\omega)\hat{C}_1(\omega)}{P_x(\omega) + P_y(\omega)}, \tilde{C}_2(\omega) = \frac{P_y(\omega)\hat{C}_2(\omega)}{P_x(\omega) + P_y(\omega)} \tag{2}$$

where $P_x(\omega)$ and $P_y(\omega)$ are $\int\int \|\phi_x(x,y,\omega)\|^2 dxdy$ and $\int\int \|\phi_y(x,y,\omega)\|^2 dxdy$ respectively. As the surface $\tilde{Z}$ is the inverse transform of $\tilde{C}(\omega)$, then $\tilde{Z}_1$ and $\tilde{Z}_2$ are the inverse transforms of $\tilde{C}_1(\omega)$ and $\tilde{C}_2(\omega)$ respectively, and $\tilde{Z} = \tilde{Z}_1 + \tilde{Z}_2$.

In the rightmost column of Figure 1(a) we show as intensity plots, the contributions to the height maps from $\tilde{Z}_1$ (top) and $\tilde{Z}_2$ (bottom).

In a complementary manner, depth maps corresponding to each representation are shown in Figure 1(b). From left to right we show: the x and y components of the surface gradient representation (top row), the x and y components of the Fourier representation (middle row) and height representation (bottom row).

3 Construction of the Models

The face database used for constructing the surface models was provided by the Max-Planck Institute for Biological Cybernetics in Tuebingen, Germany. As described in [7], this database was constructed using Laser scans ($Cyberware^{TM}$) of 200 heads of young adults, and provides head structure data in a cylindrical representation. For constructing the height based model, we converted the cylindrical coordinates to Cartesian coordinates and solved for the values of $Z(x,y)$. We were also provided with the ground-truth surface normals for the faces, which we used to calculate the surface gradient and Fourier domain eigenmodes.

For model construction, instead of using intensity images to generate the eigenmodes, as with eigenfaces, we used the eigenmodes obtained from height, surface gradient and Fourier domain, which we will refer to as $\mathbf{\Psi}^{hgt}$, $\mathbf{\Psi}^{grd}$ and $\mathbf{\Psi}^{FB}$ respectively.

For $\mathbf{\Psi}^{hgt}$, only one set of eigenmodes is required. We used 150 examples of face height data. This model is constructed so as to reproduce the face-height data variations as a linear combination of $\mathbf{\Psi}^{hgt}$,

$$E^{hgt} = \bar{\mathbf{x}}^{hgt} + \sum_{i=1}^{l} \alpha_i \mathbf{\Psi}_i^{hgt} \tag{3}$$

To train $\mathbf{\Psi}^{grd}$, we used 150 examples of ground truth face surface gradient data. It is important to note that this model is composed of two sets of eigenmodes, constructed using the first and second partial derivatives of the training data. This model re-produces surface gradient data variations as a linear combination of the set $\mathbf{\Psi}^{grd} = \{\mathbf{\Psi}^{grd^P}, \mathbf{\Psi}^{grd^Q}\}$, with the two components:

$$E^{grd^P} = \bar{\mathbf{x}}^{grd^P} + \sum_{i=1}^{l} \alpha_i^P \mathbf{\Psi}_i^{grd^P} \quad \text{and} \quad E^{grd^Q} = \bar{\mathbf{x}}^{grd^Q} + \sum_{i=1}^{l} \alpha_i^{grd^Q} \mathbf{\Psi}_i^{grd^Q} \quad (4)$$

In the case of $\mathbf{\Psi}^{grd}$, rather than directly applying the method to height data, we used surface gradient based information to construct the eigenmodes and calculated estimates of the first and second partial derivatives, p and q. We performed integration on the the gradient field to recover surface height data.

Similarly, the Fourier domain based set of eigenmodes, $\mathbf{\Psi}^{FB}$ can also be divided into two sub-models $\mathbf{\Psi}^{FB} = \{\mathbf{\Psi}^{FB^X}, \mathbf{\Psi}^{FB^Y}\}$

$$E^{FB^X} = \bar{\mathbf{x}}^{FB^X} + \sum_{i=1}^{l} \alpha_i^X \mathbf{\Psi}_i^{FB^X} \quad \text{and} \quad E^{FB^Y} = \bar{\mathbf{x}}^{FB^Y} + \sum_{i=1}^{l} \alpha_i^{FB^Y} \mathbf{\Psi}_i^{FB^Y} \quad (5)$$

To construct the models, we used 150 examples of ground truth face surface gradient data and converted them to the surfaces $\tilde{Z}_1$ and $\tilde{Z}_2$ (as described in Section 2) through Equation 2 using Frankot and Chellappa's Fourier domain integration method. Therefore $\mathbf{\Psi}^{FB}$ characterizes the Fourier-basis surface variation as a linear combination of its eigenmodes. Given new examples, we can calculate the separate variations in $\tilde{Z}_1$ and $\tilde{Z}_2$ and add them together with their respective mean shapes to generate new surfaces of faces.

The generalization of the models, or the ability of the models to capture the features of the database from which they were constucted, is illustrated in Figure 2 (left), where the percentage of generalization is shown as a function of the number of modes used. The required number of modes was calculated through the formulae $\sum_{i=1}^{t} \lambda_i \geq f_v V_T$, where λ_i are the eigenvalues of the matrix $\mathbf{L}$, V_T is the total variance (i.e. the sum of all the eigenvalues) and f_v defines the proportion of the total variation captured by the model. We can see that $\mathbf{\Psi}^{FB}$ (solid and dotted lines) generalizes in a similar manner to $\mathbf{\Psi}^{hgt}$ (long-dashed line), both achieve more than 90% using just 20 modes. By contrast, $\mathbf{\Psi}^{grad}$ (short-dashed and dot-dashed lines) required a considerably larger number of modes (at least 100) to achieve the 90% level. Interestingly, for both $\mathbf{\Psi}^{FB}$ and $\mathbf{\Psi}^{grd}$ the x-component of the model shows a slightly better generalization than the y-component of the model. This may be attributable to the left-to-right symmetry of human faces.

4 Coefficient Adjustment Using Data-Driven Search

The next series of experiments used as input the 50 prototype intensity images (orthogonally illuminated, without texture) of the out-of-sample faces not

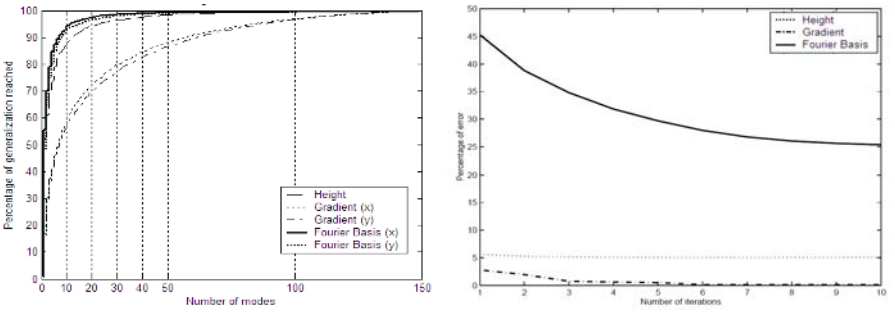


Fig. 2. Generalization of the models (left): note how $\mathbf{\Psi}^{FB}$ (solid and dotted lines) and $\mathbf{\Psi}^{hgt}$ (long-dashed line) show a similar behavior (achieving more than 90% when using at least 20 modes), while $\mathbf{\Psi}^{grad}$ (short-dashed and dot-dashed lines) requires at least 100 modes to achieve the 90% level. On the other hand, the performance tests (right) show that $\mathbf{\Psi}^{grd}$ (reaching an error of around 0.1%) outperforms both $\mathbf{\Psi}^{hgt}$ (around 5% as minimum) and $\mathbf{\Psi}^{FB}$ (around 25% as minimum).

used in training. The surface normals for these images are extracted using the geometric SFS approach described by [8], which forces the surface normals to comply with the image irradiance equation (i.e Lambert's reflectance law) as a hard constraint. We do this in the following iterative fashion:

1. Extract initial surface normals from the intensity data using SFS and transform the set of normals to one of the three Cartesian representations [2] for each of the representations (height, gradient, Fourier domain).
2. Calculate the corresponding set of coefficients α_i and back-project them back to the representation domain.
3. Re-apply Geometric SFS to this back-projected data.
4. Enforce integrability on this set of normals using Frankot and Chellappa's global integration method, described in Section 2 and return to step 1 until a desired number of iterations is reached.

Hence, instead of searching for valid linear combinations of the set of coefficients α_i, we are exploring a rather intuitive and straightforward way for adjusting their values. In this way we are also testing the flexibility of each of the models, since the input data is far from a valid model example. Figure 2 (bottom row) shows the fractional error as a function of the number of iterations. The diagram shows that $\mathbf{\Psi}^{grd}$ (reaching an error of around 0.1%) clearly outperforms both $\mathbf{\Psi}^{hgt}$ (around 5% as minimum) and $\mathbf{\Psi}^{FB}$ (around 25% as minimum). These results suggest that $\mathbf{\Psi}^{grd}$ is more robust to poor initial surface normal estimates

[2] It is important to remark that SFS information will greatly differ from the data used in the former set of experiments (though out-of-sample, the 50 faces did represent 3D face shape information), since neither integrated height nor surface gradient nor Fourier basis information derived from SFS will present the accuracy of the database examples.

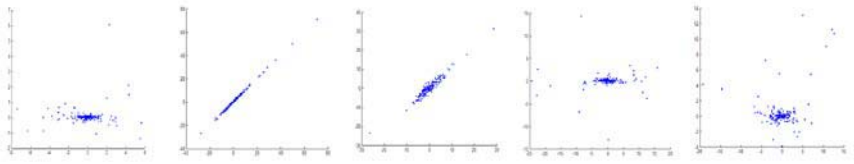


Fig. 3. Scatter plot diagrams, 10th iteration against true data. From left to right: $\mathbf{\Psi}^{hgt}$,$\mathbf{\Psi}^{grd^P}$,$\mathbf{\Psi}^{grd^Q}$, $\mathbf{\Psi}^{FB^X}$,$\mathbf{\Psi}^{FB^Y}$. Note how the surface gradient based representation is the only one that seems to be benefited from the geometric SFS adjustment.

than the alternative representations, which might need a different initialization and probably a heuristic search for the values of their coefficients. Building the eigenmodes with surface gradients works significantly better. This corroborates the choice of Sirovich's method in conjunction with image-like bases to construct models (surface gradient is closer to intensity images than height).

The former observation is reinforced by the scatter plot diagrams shown in Figure 3. The x-axis represents the shape coefficients extracted from the surface normals extracted from intensity data using 10 iterations of SFS, while the y-axis represents the shape-coefficients extracted from the surface normals estimated from the ground-truth range data. From left to right the plots show the results obtained with $\mathbf{\Psi}^{hgt}$,$\mathbf{\Psi}^{grd^P}$,$\mathbf{\Psi}^{grd^Q}$, $\mathbf{\Psi}^{FB^X}$,$\mathbf{\Psi}^{FB^Y}$. In the second and third diagrams, it is clear that the data-driven adjustment influences the correction of the initial set of shape coefficients for Ψ^{grd}, since the points in the scatter plots produce a diagonal regression-line. This regression-line is better defined for the coefficients α^{grd^P} (second image) than for α^{grd^Q} (third image), and this again suggests the models are better suited to capture left-to-right facial features (symmetry). For the remaining cases, α^{hgt} (first diagram) and α^{FB} (last two diagrams), there seems to be no relation between the geometric SFS coefficient adjustment and the true data-based shape coefficients.

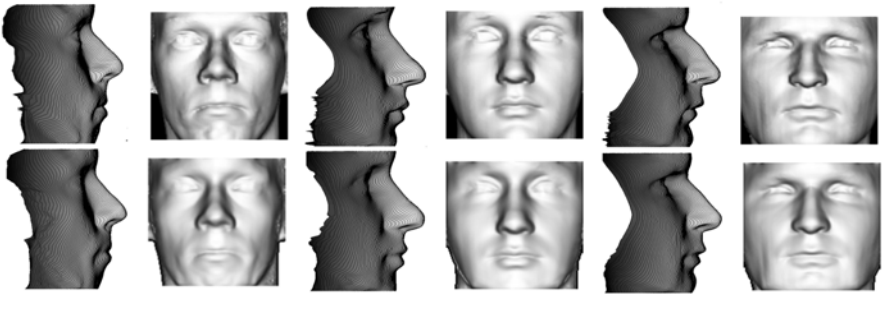


Fig. 4. Profile views of the ground truth surfaces (top row, odd columns) and gradient-based model recovered surfaces (bottom row, odd columns) after the tenth iteration. Frontal re-illumination for each surface are shown in the even columns.

Finally, Figure 4 shows profile views of the ground truth surface (top row, odd columns) and the surface recovered using the gradient-based model after the tenth iteration (bottom row, odd columns). Frontal re-illuminations corresponding to each surface are also shown in the even columns. The difference between the ground truth and the recovered surface using $\mathbf{\Psi}^{grd}$ is almost imperceptible, confirming the robustness of $\mathbf{\Psi}^{grd}$.

5 Conclusions

We have presented an analysis of three Cartesian representations for constructing three-dimensional statistical models of faces. These models were tested on SFS information, but only the surface gradient-based representation showed the necessary robustness to avoid the implementation of a specialized and time-consuming shape coefficient search technique, as required by cylindrical representations. When used to fit the model to intensity data using the image irradiance equation to constrain the surface normal direction, and also imposing integrability constraints on the recovered surface, then this representation leads to a natural way of refining the shape-coefficients. Hence we have demonstrated that 3D statistical models of faces based on Cartesian representations can work accurately without special search methods. As future work we are planning to explore the behavior of the Cartesian representations with alternative methods for shape coefficient adjustment.

References

1. Atick, J., Griffin, P. and Redlich, N. (1996), Statistical Approach to Shape from Shading: Reconstruction of Three-Dimensional Face Surfaces from Single Two-Dimensional Images, *Neural Computation*, Vol. 8, pp. 1321-1340.
2. Dovgard, R. and Basri, R. (2004), Statistical symmetric shape from shading for 3D structure recovery of faces, *European Conf. on Computer Vision (ECCV 04)*, Prague, May 2004.
3. Frankot, R.T. and Chellapa, R. (1988), A Method for Enforcing Integrability in Shape from Shading Algorithms, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, pp. 438 - 451.
4. B.K.P. Horn. (1997), Understanding Image Intensities. *Artificial Intelligence*, Vol. 8, pp. 201-231.
5. Sirovich, L. and Everson, Richard. (1992), Management and Analysis of Large Scientific Datasets,*The International Journal of Supercomputer Applications*, Vol. 6, No. 1, pp. 50 - 68.
6. Turk, M.A. and Pentland, A.P. (1991), Face Recognition Using Eigenfaces,*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586 - 591.
7. Blanz, V. and Vetter, T. (1999), A Morphable model for the synthesis of 3D Faces,*Proceedings of SIGGRAPH '99*, pp. 187 - 194.
8. Worthington, P. L. and Hancock, E. R. (1999), New Constraints on Data-closeness and Needle Map Consistency for Shape-from-shading, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 12, pp. 1250-1267.

Algorithms for Detecting Clusters of Microcalcifications in Mammograms

Claudio Marrocco, Mario Molinara, and Francesco Tortorella

Dipartimento di Automazione, Elettromagnetismo,
Ingegneria dell'Informazione e Matematica Industriale
Università degli Studi di Cassino
03043 Cassino (FR), Italy
{c.marrocco, m.molinara, tortorella}@unicas.it

Abstract. Mammography is a not invasive diagnostic technique widely used for early cancer detection in women breast. A particularly significant clue of such disease is the presence of clusters of microcalcifications. The automatic detection of such clusters is a very difficult task because of the small size of the microcalcifications and of the poor quality of the digital mammograms. In literature, all the proposed method for the automatic detection focus on the single microcalcification. In this paper, an approach that moves the final decision on the regions identified by the segmentation in the phase of clustering is proposed. To this aim, the output of a classifier on the single microcalcifications is used as input data in different clustering algorithms which produce the final decision. The approach has been successfully tested on a standard database of 40 mammographic images, publicly available.

1 Introduction

At present, mammography is the only radiological screening technique for detecting lesions in the breast using low doses of radiation. Moreover, it represents the only not invasive diagnostic technique that allows the diagnosis of a breast cancer at a very early stage, when it is still possible to successfully attack the disease with a suitable therapy. For this reason, for the female population at risk programs of wide mass screening via mammography have been carried out in many countries. One important clue of breast cancer is the presence of clustered microcalcifications. Microcalcifications are tiny granule-like deposits of calcium that appear on the mammogram as small bright spots. Their size ranges from about 0.1 mm to 0.7 mm, while their shape is sometimes irregular. Isolated microcalcifications are not, in most cases, clinically significant. However, it is difficult for radiologists to provide both accurate and uniform evaluation for the enormous number of mammograms generated in widespread screening. In this case, a computer aided analysis could be very useful to the radiologist both for prompting suspect cases and for helping in the diagnostic decision as a "second reading". The goal is twofold [1]: to improve both the *sensitivity* of the diagnosis, i.e. the accuracy in recognizing all the actual clusters and its *specificity*, i.e. the ability to avoid erroneous detections.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 884 – 891, 2005.

In the recent past, many approaches have been proposed for the automatic detection of clusters of microcalcifications [2-5], but all these methods focus on the analysis of the single microcalcifications. Generally speaking, all automatic systems are based on a two-step process: the first one is the segmentation of the mammogram while, in the second step, features derived from the segmentation are used as a basis for classifying each object as a microcalcification or as an artifact. Then, only in a following phase the filtered regions are clustered with very simple rules based on proximity of the microcalcifications, to individuate the clusters that are important for the diagnosis. However, the poor contrast on the mammogram makes microcalcifications not easily distinguishable from the mammal tissue in the background. These problems make the following feature extraction phase very critical and, moreover, errors on features will strongly affect the classification phase. As a consequence, the results of the classification can not be sufficiently reliable and the following clustering, even though very simple, can produce unsatisfactory results.

In our work we propose an alternative approach that, avoiding a decision on the single regions during the classification phase, considers the output of the classifier as a confidence degree of the single microcalcifications to be used in a successive phase of clustering. In this way, it is possible to create a suitable algorithm, that, reckoning with the spatial coordinates of the regions and the confidence degree, gives a possible partition in clusters of microcalcifications. In this way, the decision is taken on the whole cluster according to the characteristics of regions that it groups, thus leading to more reliable results. The confidence degree is so considered as an input feature for the clustering algorithm as it includes the knowledge obtained on all the extracted features in a set of non homogeneous data that represent (or are related to) the estimate of the a posteriori probability of the classifier.

In the rest of the paper we show, after a short description of the main clustering algorithms, how to adapt them to the particular situation of microcalcifications according to our method. A conclusive section describes the results obtained from experiments performed on real datasets and a comparison among the implemented algorithms.

2 Basic Clustering Algorithms

Clustering is a well known topic in the image processing field. Roughly speaking, clustering's goal is to achieve the best partition over a set of objects in terms of similarity. Similarity-base clustering is a simple technique that uses a similarity measure to guarantee if two objects are similar enough to put them in the same cluster. The similarity measure is usually defined through features of the object.

Clustering algorithms can be divided into two different categories according to the a priori knowledge on the problem that is analyzed: a first category (such as k-means) where it has to be specified the number of searched clusters and a second one where the number of clusters is variable according to the input data. Of course, the latter is more useful for our goal since we do not know anything about the presence of a cancer in the mammogram we are analyzing. Particularly, we focus on two different types of algorithms: hierarchical and sequential algorithms.

First of all, it could be useful to give the definition of clustering [6]. Given a data set $X=\{x_1,x_2,\ldots,x_N\}$ with $x_i \in R^l$, we define an *m-clustering* $\Re$ of X, the partition of X in *m* clusters, $C_1,\ldots,C_m$, so as the following conditions are satisfied:

$$\begin{cases} C_i \neq 0, \quad i=1,\ldots,m \\ \cup_{i=1}^{m} C_i = \mathbf{X} \\ C_i \cap C_j = 0, \quad i \neq j, \quad i,j=1,\ldots,m \end{cases} \tag{1}$$

In the algorithms for *hierarchical clustering,* a sequence of N nested clustering $\Re_0 \subset \Re_1 \subset \ldots \subset \Re_{N-1}$ is produced where the index of the particular element in the sequence is said the *level* in the clustering sequence. In the *agglomerative hierarchical clustering*, any pair of samples grouped together at level k remain grouped together at all higher levels. In other words, two clusters C_p and C_q contained in clustering $\Re_k$ are merged together in a single cluster present in $\Re_{k+1}$; merging is realized according to a function $g(C_i,C_j)$ that is a similarity or dissimilarity measure between clusters. Each level is then characterized by means of an *inconsistency coefficient* [6] which measures the dissimilarity of clusters that are merged. Depending on the requirements of the application at hand, a threshold is set on the inconsistency coefficient so as to choose the level containing the most "natural" number of clusters.

A second approach is given by the *sequential algorithms*, in which the samples are sequentially considered and inserted in one of the clusters. Whereas hierarchical clustering algorithms must have all data present before clustering begins, in this type of algorithms the clustering can be performed even before the full data are present. The main drawback is that clustering depends strongly on the order of data presentation. One example of this approach is the *leader clustering* [7]: given a set of samples, we consider the first sample as the centre of a circular cluster of a specified radius. Then, if the second point falls in this region it belongs to the first cluster otherwise it is the centre of a new cluster. The same for the third point and so on until all the data have been considered. An useful variation to this approach is the *leader-follower clustering* [7] that consists of changing the cluster centre when new patterns enter in a cluster. In such algorithms, the properties of the produced clusters depend on the value chosen for the distance threshold (i.e. the radius of the circle): a large threshold leads to a small number of large clusters, while a small threshold to a large number of small clusters.

3 Clustering Algorithms for Microcalcifications

The purpose of this work is to study how to find clusters of microcalcifications in mammographic images. To this aim, the clustering algorithms described in section 2 have to be adapted to the particular problem we face. As previously said, features used in the clustering algorithms are the spatial coordinates (x,y) and the confidence degree S.

3.1 Hierarchical Approach

In order to use a *hierarchical algorithm (HC)*, the idea is to consider the estimate S as a third coordinate in a three-dimensional space. Hence, the clustering is made on a

feature vector of three elements to group not only points that are near (according to a similarity measure) in the x,y plane but also on the third axis z=S.

In this way, for any pair of points $t_i = (x_i, y_i, z_i)$ and $t_j = (x_j, y_j, z_j)$ it is possible to define as similarity measure a weighted Euclidean distance:

$$d(t_i, t_j) = \sqrt{\alpha\left[(x_i - x_j)^2 + (y_i - y_j)^2\right] + \beta(z_i - z_j)^2} \tag{2}$$

where α and β are the weights used to normalize different data in a common range.

Then, using this similarity measure, it is possible to create a hierarchy of possible clustering and also to represent it with a dendrogram. The choice of the clustering is made using the inconsistency coefficient described in the previous section. When a clustering is chosen, it is possible to note that false clusters (i.e. clusters created by false regions) have a low medium value of the z component while this value is high for clusters created by true regions.

The choice of the weights is a critical point of this algorithm because we do not know how to relate the axes scales. Using the confidence degree we avoided a more problematic trouble, i.e. how to normalize all the different features; actually, we used only two weights, one for the spatial coordinates and another for the confidence degree but, referring to the sequential algorithms, it is possible to do not consider any weights.

3.2 Sequential Approach

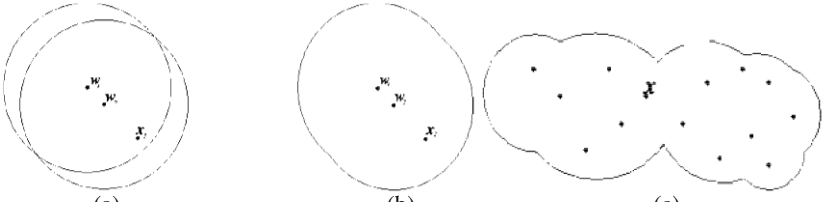
The *leader clustering (LC)*, described in the previous section, needs to choose an order for the input data, because the clustering will strongly depend on it. To this aim, a possible approach is to order the points according to the confidence degree since regions that have a higher confidence degree can more probably represent microcalcifications. So, ordering data according to decreasing values of S, the first points that the algorithm will consider are those with a higher probability of being microcalcifications and, therefore, those with a higher probability of being grouped together.

Some limits of this algorithm are that it always considers a circular form for the cluster (and this is not true in the reality) and, moreover, that it is not sure that the region with a higher S is the centre of the cluster. To overcome these problems, in the previous section the leader-follower clustering has been described. Now, we propose a different approach, called *Moving Leader (MLC)*, that consists of considering as centre of the cluster a weighted centre of mass that has to be calculated each time a new region is grouped in the cluster. The used weight is the respective confidence degree S_i. Let us consider some points of coordinates $\mathbf{x}_1,\ldots,\mathbf{x}_{n-1}$ with confidence degree respectively $s_1,\ldots s_{n-1}$ grouped in a cluster; now, if we consider a sample $\mathbf{x_n}$ with confidence degree s_n that has to be grouped in that cluster, then, the new centre will be:

$$\boldsymbol{w_n} = \frac{\sum_{i=1}^{n} s_i \boldsymbol{x_i}}{\sum_{i=1}^{n} s_i} = \frac{\sum_{i=1}^{n-1} s_i \boldsymbol{x_i} + s_n \boldsymbol{x_n}}{\sum_{i=1}^{n} s_i} = \frac{\sum_{i=1}^{n-1} s_i}{\sum_{i=1}^{n} s_i}\left(\frac{\sum_{i=1}^{n-1} s_i \boldsymbol{x_i}}{\sum_{i=1}^{n-1} s_i} + \frac{s_n}{\sum_{i=1}^{n-1} s_i}\boldsymbol{x_n}\right) \tag{3}$$

In this way, the centre of the cluster moves towards the direction where the points are more dense and so, where there is a higher probability of finding new microcalcifications. Moreover, the shape of the new cluster is not a circle but it is created merg-

ing n circles. For example in fig. 1.a, the point w_1 is the centre of a cluster, when x_2 has to be grouped in the cluster, the centre moves to w_2. The shape of the region that contains the final cluster is the union between the two circles (fig. 1.b). When a lot of points are analyzed we have a shape like in fig. 1.c.



(a) (b) (c)

Fig. 1. (a) w_1 is the centre of a cluster, when x_2 has to be grouped in the cluster, the centre moves to w_2 **(b)** the shape of the cluster is the union of two circles **(c)** an example of final shape of the cluster

When clusters have been created, a post-processing operation can be performed to better group microcalcifications. This strategy consists of merging two clusters that share at least one microcalcification. This approach, that we called *Moving Leader with Merge (MLMC)*, let us join near regions avoiding an excessive number of clusters that is not corresponding to the reality.

4 Experimental Results

The system has been tested on a standard database provided by courtesy of the National Expert and Training Centre for Breast Cancer Screening and the Department of Radiology at the University of Nijmegen, the Netherlands. It contains 40 digitized mammographic images composed of both oblique and craniocaudal views from 21 patients. All images have a size of 2048x2048 pixels and use 12 bits per pixel for the gray levels. Each mammogram has one or more clusters of microcalcifications marked by radiologists; each cluster is described by the centre and the radius of a circle totally containing it. The total number of clusters is 105, 76 of which are malignant and 29 benign.

As in [8], segmentation by Tree-Structured Markov Random Field (TS-MRF) and two-stage classification has been performed. Thus, for each region, we have at our disposal the coordinates and the output of an SVM classifier that indicates the signed distance of a region from the optimal separating hyperplane between the true and the false class. Using these input data, the clustering algorithms described in section 3 can be applied. For each algorithm, five different runs were executed according to an opportune parameter; for the hierarchical algorithm, the inconsistency coefficient has been varied from the 75% to the 95% of its maximum value with a 5% step, while for the sequential algorithms, a mean radius R_m has been evaluated from all the clusters marked by radiologists and the experiments were performed using R_m plus or minus the 10% and the 20% of its value.

The performance of the proposed algorithms were only evaluated on the true positive clusters since we want to show how our method fits with the description of microcalcifications clusters. To highlight that, a dimensionless parameter, that we called *covering factor (CF)*, has been introduced; this takes into account if a cluster is not well recognized, i.e. if a cluster of the ground truth is recognized but it is subdivided in more than one sub-cluster or if an identified cluster is greater than the corresponding circle in the ground truth. For the *j-th* cluster in the ground truth, we have:

$$CF_j = \frac{\sum_{k=1}^{N_j} W_{jk} I_{jk}}{G_j} \tag{4}$$

where N_j is the number of sub-clusters, I_{jk} is the area of the intersection between the *k-th* sub-cluster and the *j-th* circle in the ground truth, G_j is the area of that circle and $W_{jk} = I_{jk} / (N_j * A_{jk})$, with A_{jk} area of the *k-th* sub-cluster, is a weight that reaches its maximum value, i.e. one, when there is only one sub-cluster totally contained in the circle of the ground truth. Fig.2 shows how to evaluate the CF for a cluster subdivided into three sub-clusters, two totally contained in the circle of the ground truth and one that trespasses the boundary of the ground truth.

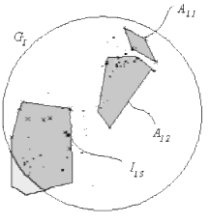
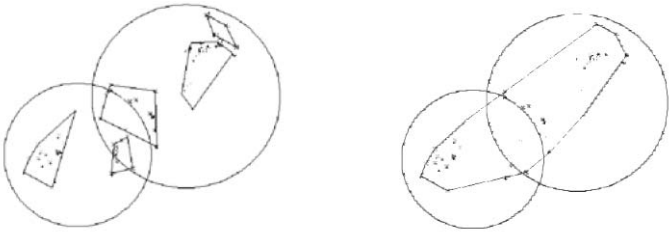


Fig. 2. Evaluation of the covering factor: in this case, $CF_1 = (W_{11}I_{11}+W_{12}I_{12}+W_{13}I_{13})/G_1$ where $I_{11}=A_{11}$, $I_{12}=A_{12}$ and $W_{11}=I_{11}/3A_{11}=1/3$, $W_{12}=I_{12}/3A_{12}=1/3$,$W_{13}=I_{13}/3A_{13}$.

However, we are comparing the cluster area evaluated by our method (that is based on the convex hull of regions) with the circles made by radiologists that are not very accurate. As a consequence, the CF never reaches high values, i.e. it is impossible that the estimated area is equal to the area of the circle in the ground truth. If this happens, it is due to a bad behaviour of the clustering algorithms that groups too big regions; this is especially visible when the inconsistency coefficient for the hierarchical clustering or the medium radius for the sequential algorithms becomes too great. An example is presented in fig. 3: fig. 3.a shows the best situation for a hierarchical algorithm with inconsistency coefficient equal to 85% while in fig. 3.b we can see that a cluster is lost for an inconsistency coefficient equal to 95%.

Thus, to better evaluate the performance of the algorithms, we also have to look at the number of clusters detected. Hence, we consider an algorithm better than another when, for the same number of detected clusters, it has the greater CF. The results obtained in terms of medium CF over all the images and percentage of detected clusters are reported in table 1.a for the hierarchical algorithm and in table 1.b for the sequential algorithms.



(a) **(b)**

Fig. 3. Clustering of image c04c with a hierarchical algorithm: **(a)** with an inconsistency coefficient equal to 85% the two clusters are correctly recognized while **(b)** a cluster is lost with an inconsistency coefficient equal to 95%

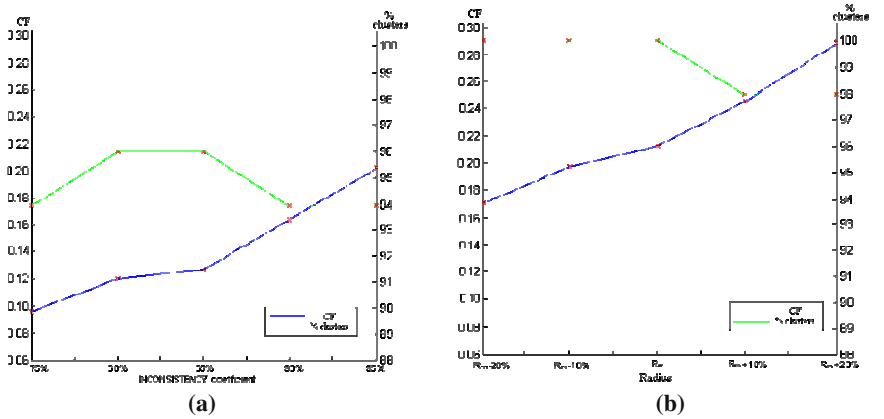
Table 1. Results obtained in terms of CF and percentage of detected clusters for the hierarchical **(a)** and the sequential **(b)** algorithms.

(a)

	HC	
IC	**CF**	**% cl.**
75%	0.09	94%
80%	0.11	96%
85%	0.12	96%
90%	0.15	94%
95%	0.19	94%

(b)

	LC		MLC		MLMC	
R	**CF**	**%cl.**	**CF**	**%cl.**	**CF**	**%cl.**
R_m-20%	0.09	98%	0.11	98%	0.17	100%
R_m-10%	0.12	100%	0.13	100%	0.20	100%
R_m	0.14	100%	0.15	100%	0.21	100%
R_m+10%	0.15	100%	0.16	98%	0.24	98%
R_m+20%	0.18	98%	0.19	96%	0.28	98%



(a) **(b)**

Fig. 4. Graphical results obtained in terms of CF and percentage of detected clusters as a function of the IC for the HC **(a)** and of the radius for the best sequential algorithm, i.e. the MLMC **(b)**

Then, in fig.4 we report a graph of these results with two axes, one for the CF and one for the percentage of detected clusters. Fig. 4.a shows that the hierarchical algorithm never reaches a percentage of detection equal to 100%, while in fig. 4.b

we show the results for the MLMC that outperforms the other sequential algorithms. In fact, it reaches the 100% of detected clusters with a CF greater than all the other algorithms.

In summary, the experiments show that it is possible to improve the clustering of microcalcifications clusters, especially with the MLMC algorithm. It remains to test the algorithms on the clustering of the false regions, i.e. to demonstrate that the distribution of false positive does not influence the good capability of clustering shown by these algorithms.

References

1. De Santo, M., Molinara, M., Tortorella, F., Vento, M.: Automatic Classification of Clustered Microcalcifications by a Multiple Expert System. Pattern Recognition 36 (2003), 1467 – 1477.
2. Karssemeijer, N.: Adaptive Noise Equalization and Recognition of Microcalcification Clusters in Mammograms. International Journal of Pattern Recognition and Artificial Intelligence 7 (1993), 1357 1376.
3. Strickland, R.N., Hahn, H.: Wavelet Transforms for Detecting Microcalcifications in Mammograms. IEEE Transaction on Medical Imaging 15 (1996), 218–229.
4. Netsch, T., Peitgen, H.: Scale Space Signatures for the Detection of Clustered Microcalcifications in Digital Mammograms. IEEE Transaction on Medical Imaging 18, (1999), 774 786.
5. Cheng, H.D., Lui, Y.M., Freimanis, R.I.: A Novel Approach to Microcalcification Detection Using Fuzzy Logic Technique. IEEE Transaction on Medical Imaging 17 (1998), 442–450.
6. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 2nd Edition. Elsevier Science (2003).
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd Edition. John Wiley & Sons, Inc. (2001).
8. D'Elia, C., Marrocco, C., Molinara, M., Poggi, G., Scarpa, G., Tortorella, F.: Detection of Microcalcifications Clusters in Mammograms through TS-MRF Segmentation and SVM-based Classification. Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, IEEE Computer Society Press (2004).

Clustering Improvement for Electrocardiographic Signals

Pau Micó[1], David Cuesta[1], and Daniel Novák[2]

[1] Department of Systems Informatics and Computers, Polytechnic School of Alcoi, Plaza Ferràndiz i Carbonell 2, 03801 Alcoi, Spain
{pabmitor, dcuesta}@disca.upv.es
[2] Department of Cybernetics, Czech Technical University in Prague, Czech Republic
novakd@bio.felk.cvut.cz

Abstract. Holter signals are ambulatory long-term electrocardiographic (ECG) registers used to detect heart diseases which are difficult to find in normal ECG. These signals normally include several channels and its duration is up to 48 hours. The principal problem for the cardiologists consists of the manual inspection of the whole Holter ECG to find all those beats whose morphology differ from the normal cardiac rhythm. The later analysis of these abnormal beats yields a diagnostic from the pacient's heart condition. In this paper we compare the performance among several clustering methods applied over the beats processed by Principal Component Analysis (PCA). Moreover, an outlier removing stage is added, and a cluster estimation method is included. Quality measurements, based on ECG labels from MIT-BIH database, are developed too. At the end, some results-accuracy values among several clustering algorithms is presented.

1 Introduction

The development and improvement of biosignal recording devices implies a quality increase of the acquired signals that becomes an important problem both in the storage and on the processing. Furthermore, the big amount of information is obtained from this kind of records. In this paper, we are going to deal with long-term Holter Electrocardiographic signals applying, firstly a compressing stage in order to reduce the signal size and, secondly, several clustering techniques with the goal of making an ECG analysis as easy as possible and optimizing time-processing performance. Therefore,by means of the clustering task the number of beats is reduced, faciliting the cardiologist's diagnosis. Along this paper we present the compressing task developed by using a polygonal approximation of the original ECG, and an improvement of the clustering task as the outlier removing stage. In addition, a comparative study among several well-known clustering algorithms for this specific ECG clustering task improvement is carried out. Next, the results validation step is presented using the quality measures proposed in *Section 2.3*. Finally the conclusions are presented.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 892–899, 2005.

2 Methodology

The methodology followed to enhance the Holter clustering task is summarized in *Figure 1.*

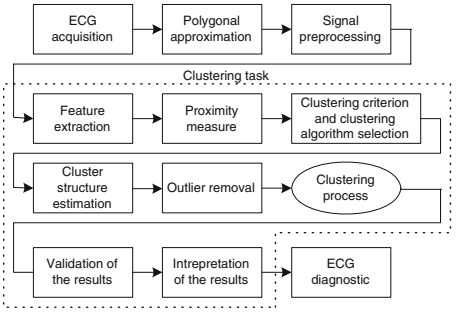


Fig. 1. Basic steps for the whole clustering Holter ECG process.

2.1 Polygonal Approximation

To further alleviate the computational burden in later processing steps, it is necessary to simplify the Holter by using any kind of polygonal approximation process [1]. For solving both, the storage-space and the time-processing problems, a comparative study between polygonal algorithms and metrics has been performed [2].

2.2 Signal Preprocessing

In most of the cases a 24 hour Holter recording is composed by more than 110000 beats. Trying to alleviate the computational burden, preprocessing tasks are developed over the compressed ECG instead of working with the original one. The compressed ECG is composed by a certain amount of polygonally approximated beats. As the signal preprocessing plays a very important role for further ECG clustering, the followins steps have been carried out: *(i) characteristic point detection and beat segmentation* [3], *(ii) baseline removal* [4] and *(iii) signal denoising*[5]. As a result of the preprocessing, we will obtain a clearly set of compressed and segmented beats.

2.3 Clustering Task

The goal of any Holter computer-aided process is to finally separate heart beats into different groups. The fact of classifying objects by non-supervised way within these groups is known as clustering task [6].

Feature Extraction. The feature extraction stage is used to facilitate the dissimilitude evaluation between objects. If the selected features does not represent the intrinsic quality of each object, the final results derived from clustering process will not become acceptable. The object feature selection can be based on many different techniques [7], [8], [9]. In this case and because of its high-speed and mathematical simplicity, the PCA has been chosen [10]. For applying this analysis to the Holter it is necessary to adapt the input objects to the PCA requirements by defining two parameters:

1. The selected ***feature***, from the beats to become the PCA data input matrix A. As the compressed beat is made by 2-dimensional segments (an amplitude sample acquired in a concrete time), we can compose the matrix by three ways: either by the amplitude samples a_i, by the time samples t_i or by using a combination of both, as the slope $(\frac{a_i}{t_i})$. Results are shown in *Figure 2*.
2. The ***number of variables*** to be extracted from the original space. In this case we have used the number of segments needed to approximate the most compressed Holter ECG beat.

Proximity Measure. This is a measure that quantifies how *similar* or *dissimilar* two feature vectors are. In order to compare and select the metric with best results, the following estimators have been tested: Euclidean, city block, correlation, Mahalanobis and cosine [6].

Clustering criterion and Algorithm Selection. The clustering criterion depends on the interpretation the expert gives to the term *sensible* based on the type of clusters that are expected to underlie the data set. It may be expressed via a cost function or some other types of rules. We will have to choice a specific algorithmic scheme that unravels the clustering structure of the data set.

The Cluster Structure Estimation Stage. In order to improve the results, we turn our attention to the task of determining the best clustering that have been tested within a given hierarchy. Clearly, this is equivalent to the identification of the number of clusters that best fits the data. In this way, we have implemented an intrinsic method where, only the structure of the data set X is taken into account [6]. According to this method, and with C_i and C_j representing two different clusters, the final clustering $\Re_t$ must satisfy the following relations:

$$d_{min}^{SS}(C_i, C_j) > \max\{h(C_i), h(C_j)\}, \quad \forall C_i, C_j \in \Re_t \tag{1}$$

$$d_{min}^{SS}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \tag{2}$$

where the function $h(C)$ measures the dissimilarity between the vectors of the same cluster C. In words, for the estimation of the number of clusters a loop is repeated since, in the final clustering, the dissimilarity between every pair of clusters is larger than the *self-similarity* of each of them given by $h(C)$.

The Outlier Removing Stage. We define as outlier corrupted object whose features hardly differ from the others, without belonging to any cluster. Depending on the clustering algorithm used, the results are more or less sensible

to the outliers. Because of this reason, we have added a removing stage to clear the data set from outliers. In this way, and taking into account the serial feature of the electrocardiographic signals (cardiologists talk about arrhythmical sequences and not about isolated arrhythmical beats) isolated beats that appear far away from the biggest cluster (in terms of similarity distance) are considered as outliers and are removed from the original data set.

Validation of the Results. Once the results have been obtained, we have to verify its correctness. Starting from a sequence of L beats, that have been finally grouped in n clusters named $\{C_1 \dots C_n\}$, where l_a means the (labeled) object belonging to the class A and T^a means the fact of classifying one object within the cluster A, the validation is carried out by means of the quality estimators proposed next:

- $P(l_a)$: is the likelihood of the beats from the class A.
- $P(l_{\overline{a}})$: is the likelihood of the beats that are not labeled as pertaining to the class A.
- $P(T^a)$: is the likelihood of classifying the beat l_i within the cluster A.
- $P(T^{\overline{a}})$: is the likelihood of classifying the beat l_i within a cluster that is not the cluster A.
- *True Positive (TP)*: is the right classsification fact (hit), including into a cluster an object that is (a priori) labeled as relevant to it.
- *True Negative (TN)*: is the right classification fact that rejects from a cluster an object that does not belong to it.
- *False Positive (FP)*: is the wrong classification fact (miss), including into a cluster an object that is not (a priori) labeled as relevant to it.
- *False Negative (FN)*: is the wrong classification fact that rejects from a cluster an object that, in fact, indeed belongs to it.

Equation 3 is used to evaluate the single accuracy obtained by a concrete cluster.

$$ACC_a = TNF \cdot P(l_{\overline{a}}) + TPF \cdot P(l_a) \tag{3}$$

We use the *Equation 4* for the total accuracy in the clustering task. An ACC_{total} value next to the unit means a good clustering task result.

$$ACC_{total} = \sum_{a=1}^{n} ACC_a \cdot P(l_a) \tag{4}$$

3 Experiments and Results

As a consequence of each one of the stages commented above we have performed a series of experiments in the aim of providing the best results. Experiments have been performed over 45 Holter ECG containing 44630 heart beats. The sources come from MIT-BIH Arrhythmia database [11]. Results are presented in the figures below where, in *Figures 4*, *5* and *6*, a dotted line has been included to reflect the real number of existent clusters. Clustering accuracy is given by *Equation 4*.

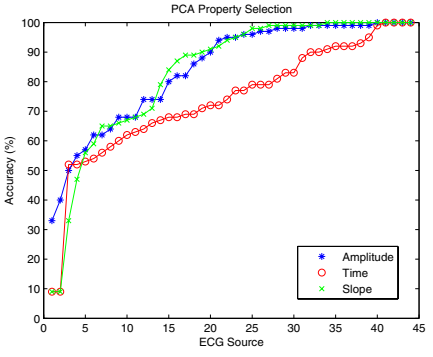


Fig. 2. Holter ECG clustering accuracy by selecting different features in PCA.

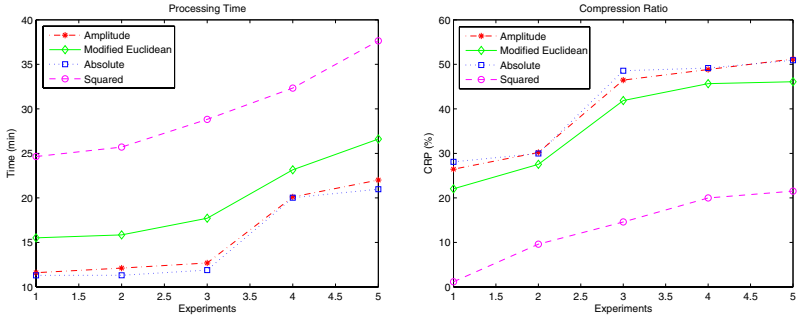


Fig. 3. Processing time (left) and compression ratio (CPR) (right) obtained in the polygonal approximation process by the use of different metrics. In the X-axe, five different source ECG signals from MIT database [11] have been used.

- ***Feature selection test.*** Where there has been compared amplitude, time and slope features in order to get the best ECG polygonal approximation (*Figure 2*).
- ***Polygonal approximation metric test.*** In order to best evaluate the polygonal approximation process, several metrics have been used: absolute, squared and amplitude error metrics and modified Euclidean distance metric. Experimental results are shown in *Figure 3*.
- ***Clustering algorithm selection test.*** Three different algorithms have been tested: *(i) K-Means* [12], *(ii) Max-Min* [13] and *(iii) Binary* [6]. Experiments performed for the best metric and algorithm selection are shown in *Figures 4* and *5*.
- ***Cluster structure estimation test.*** Results from the cluster structure estimation are shown in *Figure 6*.
- ***Outlier removing test.*** In this test, the results are given in terms of TP and FP detections [14]. Notice that it is important to minimize the FP in order not to remove a beat that is not an outlier. When no FP outliers have been detected, the best accuracy clustering results are achieved (*Table 1*).

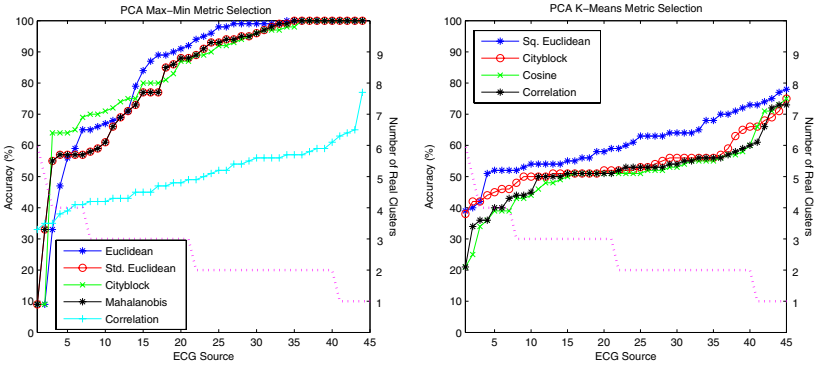


Fig. 4. Metric improvement for the K-Means (right) and the Max-Min (left) clustering algorithms using PCA feature extraction.

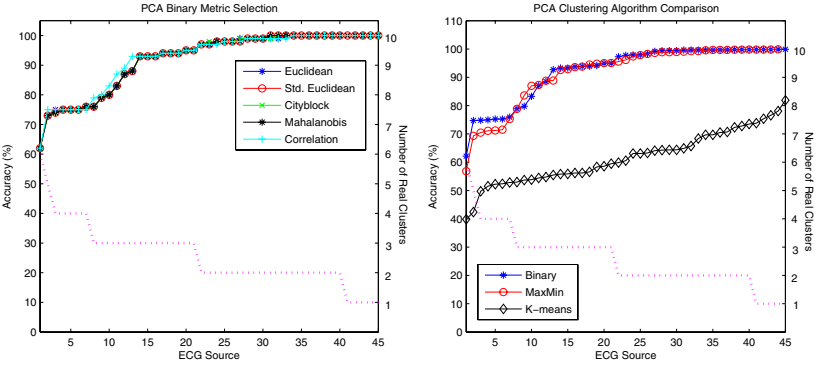


Fig. 5. (Left) Metric improvement for the Binary clustering algorithm using PCA feature extraction. (Right) Best comparative results between the three used algorithms.

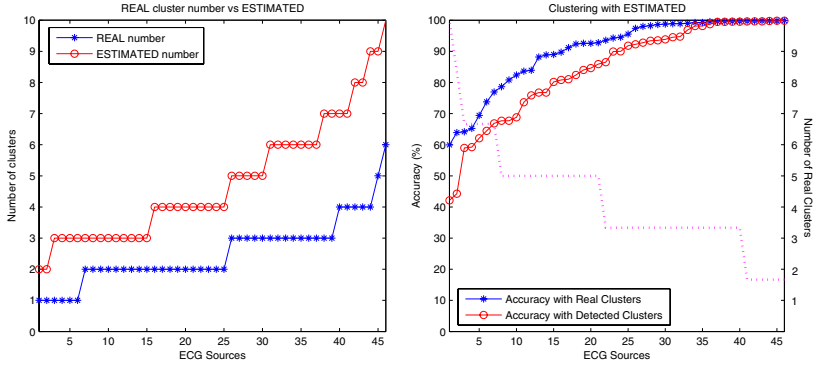


Fig. 6. (Left) We use the intrinsic method to estimate the number of clusters. (Right) The clustering task comparison using real and detected clusters.

Table 1. In the outlier removing stage a Max-Min algorithm with intrinsic number of clusters detection has been used. No FP detections gives the best accuracy.

ECG Source	TP	FP	TPF (%)	FPF (%)	Outlier detection Accuracy (%)
1002	0	0	100	0	100
1003	2	0	100	0	100
1004	2	0	100	0	100
100	2	0	33.3	0	66.6
1010	1	0	33.3	0	66.6
103	2	0	100	0	100
104	2	0	100	0	100
10	1	0	33.3	0	66.6
110	2	0	66.6	0	83.3
203	1	0	0.9	0	50.5
205	0	1	0	100	0
213	1	0	25	0	62.5
214	2	0	25	0	62.5

4 Discussion and Conclusion

Depending on the commented stage, we can withdraw the following conclusions:

For the ECG polygonal approximation stage, the best performance in terms of compression ratio and processing time yields absolute metric, that offers no critical information losses with compression ratios next to 50%.

Considering the variables for the PCA data matrix, a linear combination of time and amplitude samples gives the best result (see *Figure 2*).

The automatic cluster structure evaluation stage developed using the intrinsic becomes optimistic since over-measures the number of clusters (because of the non-detected outliers from the necessarily too conservative removing stage). Despite this fact, the accuracy obtained is only about a 10% worse that the achieved with the exact number of clusters (*Figure 6*).

In *Figures 4* and *5-(left)*, we can check that simple metrics perform the best clustering results giving us a reasonable accuracy rate, independently from the algorithm selected for the clustering task.

From the comparison between clustering algorithms (*Figure 5-(right)*) is derived that Binary and Max-Min performance is 75% for difficult clustering problems (high number of hidden clusters) and nearly 100% for the simple ones. It is not advisable to use K-Means algorithm in Holter ECG clustering tasks because of its high dependency on parameters or such as the number of clusters, the outliers or the cluster initialization.

If we interpretate the results related with the special morphology of the clusters extracted from an ECG signal we can realize how the most of normal sinus rhythm are grouped in a major cluster. On the opposite, a few beats in few clusters are presented. In addition, several outliers can appear too. As the diagnostic is made by through the abnormal beats, this fact gives relevance

to that clusters with a few beats, instead of giving it to the major cluster. Consequently, the cluster structure over-measurament becomes worthless and the very important question is to detect as much as abnormal clusters as possible.

References

1. Koski A., and Juhola M.: Segmentation of Digital Signals Based on Estimated Compression Ratio. IEEE trans. on Biomedical Engineering, Vol. 43(9), (1996)
2. Micó P., Cuesta D., and Novák D.: Polygonal Approximation of Holter Registers: A Comparative Study for Electrocardiographic Signals Time Compression. Accepted in Computational Intelligence in Medicine and Healthcare proc., (2005)
3. Cuesta, D. and Novák, D.: Automatic extraction of significant beats from a Holter register. BIOSIGNAL proceedings, pp. 3-5, (2002)
4. Cuesta D., Novák D., Eck V., Pérez C. and Andreu G.: Electrocardiogram Baseline Removal Using Wavelet Approximation. BIOSIGNAL proc., pp. 136-138, (2000)
5. Novák D., Cuesta D., Eck V., Pérez J.C. and Andreu G.: Denoising Electrocardiogram Signal Using Adaptive Wavelets. BIOSIGNAL proc., pp. 18-20, (2000)
6. Theodoridis S., and Koutroumbas K.: Pattern Recognition. Academic Press, (1999)
7. Rabiner L.R., and Juang B.H.: An Introduction to Hidden Markov Models. IEEE ASSP Magazine, (1986)
8. Micó P., Cuesta D., and Novák D.: Pre-clustering of Electrocardiographic Signals Using Ergodic Hidden Markov Models. LNCS Vol. 3138, pp. 939-947, (2004)
9. Harris R. J.: Multivariate analysis of variance. Statistics: Textbooks and monographs, Vol. 137, pp. 255-296, (1993)
10. Micó P., Cuesta D., and Novák D.: High-Speed Feature Extraction in Holter Electrocardiogram using Principal Component Analysis. BIOSIGNAL proc., (2004)
11. Mark,R., and Moody G.: MIT-BIH arrhythmia data base directory. Massachusetts Institute of Technology-Beth Israel Deaconess Medical Center, (1998)
12. González R.C., and Tou J.T.: Pattern Recognition Principles. Addison- Wesley Publishing Company, (1974)
13. Juan. A.: Optimización de Prestaciones en Técnicas de Aprendizaje No Supervisado y su Aplicación al Reconocimiento de Formas. PhD thesis, Universidad Politécnica de Valencia, (1999)
14. Rangayyan, R.M.: Biomedical Signal Analysis. A Case-Study Approach. Wiley-IEEE Press, (2002)

Mammogram Analysis Using Two-Dimensional Autoregressive Models: Sufficient or Not?

Sarah Lee[1,2,⋆] and Tania Stathaki[3]

[1] Department of Diabetes, Endocrinology & Internal Medicine,
Guy's, King's and St Thomas' School of Medicine, King's College London,
Denmark Hill Campus, New Medical School Building,
Bessemer Rd, London. SE5 9PJ UK
[2] Brain Image Analysis Unit, Centre for Neuroimaging Sciences,
Institute of Psychiatry, De Crespigny Park, London, SE5 8AF, UK
[3] Communications and Signal Processing Group,
Department of Electrical and Electronic Engineering,
Imperial College London, Exhibition Rd, London. SW7 2AZ UK
sarah.lee@iop.kcl.ac.uk, tania@imperial.ac.uk

Abstract. Two-dimensional $(2-D)$ autoregressive (AR) models have been used as one of the methods to characterise the textures of tumours in mammograms. Previously, the $2-D$ AR model coefficients were estimated for the block containing the tumour and the blocks in its 3×3 neighbourhood. In this paper, the possibility of having the estimated set of AR model coefficients of the block containing the tumour as a unique set of AR model coefficients for the entire mammogram is looked into. Based on the information given from the MiniMammography database, the possible number of blocks of the same size of the block containing the tumour is obtained from the entire mammogram and for each block a set of AR model coefficients is estimated using a method that combines both the Yule-Walker system of equations and the Yule-Walker system of equations in the third-order statistical domain. These sets of AR model coefficients are then compared. The simulation results show that 98.6% of the time we can not find another set of AR model coefficients representing the blocks of pixels in the possible neighbourhood of the entire mammogram for the data (95 mammograms with 5 of them having two tumours) available in the MiniMammography database.

1 Introduction

The two-dimensional $(2-D)$ autoregressive modelling technique has been used as one of the statistical methods to characterise and analyse the textures in images [4][2]. The images is divide into blocks of pixels and for each block a set of AR model coefficients is estimated. Alternatively, for the area of interest a set of AR model coefficients is estimated.

⋆ The author would like to thank Professors A.G. Constantinides and C. Xydeas for their suggestions and comments. This work was carried out while the author was with Communications and Signal Processing Group, Imperial College London.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 900–906, 2005.

A number of methods are available in the literature for estimating the AR model coefficients, including the Yule-Walker system of equations (YW) [1], the Yule-Walker system of equations in the third-order statistical domain (YWT) [8], the combined method using the two aforementioned methods [4], the constrained optimisation formulation with equality constraints (ConEq) [4] and the constrained optimisation formulation with inequality constraints (ConIneq) [4]. From [4], the comparison among these methods can be found for synthetic images which are built using a known stable $2-D$ AR model and using a non-Gaussian driving input. The images are also contaminated with an external Gaussian noise. The YW method is capable of estimating the AR model coefficients when the external noise is small. The YWT method can give good estimations when the external noise is large, however, the variances calculated using the estimations from a number of realisations are higher than the YW method. The rest of aforementioned methods were proposed to improve these problems.

In [2] and [4], the $2-D$ AR modelling technique is applied to mammogram analysis. The block containing the tumour was analysed and the coefficients representing the block were claimed to be symmetric. However, by taking a block of the same size of pixels in any of them in its possible neighbourhood, can we find another set of AR model coefficients which have values close to the one representing the tumour block? This is the question we try to answer in this paper.

All mammograms containing a tumour or tumours are taken from the database [5]. There are 95 of them to analyse, including 5 of mammograms containing two tumours. The combined method that employs both the Yule-Walker system of equations and the Yule-Walker system of equations in the third-order statistical domain [4] is used to estimate the AR model coefficients. The reason for choosing this method is that in [4] it was proven that this method is capable of estimating AR model coefficients in both low and high SNR environments.

The paper is organised as follows. In Section 2, the $2-D$ AR model is revisited. In Sections 3, 4 and 5, the conventional Yule-Walker system of equations, the Yule-Walker system of equations in the third-order statistical domain and the combined method can be found respectively. How the possible neighbourhood around the tumour is defined is explained in Section 6. In Section 7, simulation results can be found. The conclusion and summary are given in Section 8.

2 Two-Dimensional Autoregressive Model

Let us consider a digitised image x of size $M \times N$. Each pixel of x is characterised by its location $[m, n]$ and can be represented as $x[m, n]$, where $1 \leq m \leq M$, $1 \leq n \leq N$ and $x[m, n]$ is a positive intensity (gray level) associated with it. A two-dimensional $(2-D)$ autoregressive (AR) model is defined as [1]

$$x[m,n] = -\sum_{i=0}^{p_1}\sum_{j=0}^{p_2} a[i,j]x[m-i,n-j] + w[m,n], \tag{1}$$

where $[i, j] \neq [0, 0]$, $a[i, j]$ is the AR model coefficient, $w[m, n]$ is the input driving noise, and $p_1 \times p_2$ is the order of the model.

The driving noise, $w[m,n]$, is assumed to be zero-mean, i.e., $E\{w[m,n]\}=0$ and non-Gaussian. The AR model coefficient $a[0,0]$ is assumed to be 1 for scaling purposes, therefore we have $[(p_1+1)(p_2+1)-1]$ unknown coefficients to solve.

An external zero-mean Gaussian noise, $v[m,n]$, is added onto the system, so that the method will be able to deal with real images which are contaminated with such noise. Mathematically the new system can be written as

$$y[m,n]=x[m,n]+v[m,n]. \tag{2}$$

The signal-to-noise ratio (SNR) of the system is calculated by

$$SNR=10log_{10}\frac{\sigma_x^2}{\sigma_v^2} \quad dB \tag{3}$$

where σ_x^2 is the variance of the signal and σ_v^2 is the variance of the noise.

3 Yule-Walker System of Equations

The conventional Yule-Walker equations are given by [1][6]

$$\sum_{i=0}^{p_1}\sum_{j=0}^{p_2}a[i,j]r_{yy}[i-k,j-l]=-r_{yy}[-k,-l] \tag{4}$$

for $k=0,\cdots,p_1$ and $l=0,\cdots,p_2$, where $[k,l]\neq[0,0]$, $[i,j]\neq[0,0]$, $a[i,j]$ is the AR model coefficient, $1\leq m\leq M, 1\leq n\leq N$, $M\times N$ is the size of the given image and $r_{yy}[i,j]=E\{y[m,n]y[m+i,n+j]\}$.

Equation (4) can be written in vector-matrix form as

$$\underline{\boldsymbol{R}}\underline{\boldsymbol{a}}=-\underline{\boldsymbol{r}}, \tag{5}$$

where $\underline{\boldsymbol{R}}$ is a $(p_1p_2+p_1+p_2)\times(p_1p_2+p_1+p_2)$ matrix and $\underline{\boldsymbol{a}}$ and $\underline{\boldsymbol{r}}$ are both $(p_1p_2+p_1+p_2)\times 1$ vectors.

These equations give good AR model coefficient estimations when the SNR is high. However, the error increases with σ_v^2.

4 Yule-Walker System of Equations in the Third-Order Statistical Domain

The equations that relate the AR model parameters to the third-order moment samples are [6][8]:

$$\sum_{i=0}^{p_1}\sum_{j=0}^{p_2}a[i,j]C_{3y}([i-k,j-l],[i-k,j-l])=-C_{3y}([-k,-l],[-k,-l]) \tag{6}$$

for $k=0,\cdots,p_1$, $l=0,\cdots,p_2$ and $[k,l]\neq[0,0]$, where
$C_{3y}([i_1,j_1],[i_2,j_2])=E\{y[m,n]y[m+i_1,n+j_1]y[m+i_2,n+j_2]\}$ for zero-mean process $y[m,n]$.

These equations are insensitive to external Gaussian noise. The equations can be written in matrix form as

$$\underline{\boldsymbol{C}}\underline{\boldsymbol{a}} = -\underline{\boldsymbol{c}}, \tag{7}$$

where $\underline{\boldsymbol{C}}$ is a $(p_1p_2 + p_1 + p_2) \times (p_1p_2 + p_1 + p_2)$ matrix and $\underline{\boldsymbol{a}}$ and $\underline{\boldsymbol{c}}$ are both $(p_1p_2 + p_1 + p_2) \times 1$ vectors.

5 The Method Combining the YW and the YWT

The method using both the YW and the YWT techniques was defined as [3][4]

$$\underline{\boldsymbol{D}} \begin{pmatrix} \underline{\boldsymbol{R}} \\ \\ \underline{\boldsymbol{C}} \end{pmatrix} \underline{\boldsymbol{a}} = -\underline{\boldsymbol{D}} \begin{pmatrix} \underline{\boldsymbol{r}} \\ \\ \underline{\boldsymbol{c}} \end{pmatrix} \tag{8}$$

where the matrix $\underline{\boldsymbol{R}}$ and vector $\underline{\boldsymbol{r}}$ are defined in (5),
the matrix $\underline{\boldsymbol{C}}$ and vector $\underline{\boldsymbol{c}}$ are defined in (7),
$\underline{\boldsymbol{D}}$ is a diagonal weighting matrix, and
$\underline{\boldsymbol{a}}$ is the vector of the unknown AR model coefficients,
$[a[0,1], \cdots, a[0,p_2], \cdots, a[p_1,p_2]]^T$.
Let us consider the formulation where the system is contaminated with external Gaussian noise.

$$\underline{\boldsymbol{R}}_{yy}\underline{\boldsymbol{a}} + \underline{\boldsymbol{r}}_{yy} = \sigma_v^2 \underline{\boldsymbol{I}}\underline{\boldsymbol{a}} \tag{9}$$

where the AR model coefficients estimation $\underline{\boldsymbol{a}}$ is obtained from (7) using

$$\underline{\boldsymbol{a}} = -\underline{\boldsymbol{C}}^{-1}\underline{\boldsymbol{c}}. \tag{10}$$

Let $\underline{\boldsymbol{r}}_1 = \underline{\boldsymbol{R}}_{yy}\underline{\boldsymbol{a}} + \underline{\boldsymbol{r}}_{yy}$, where $\underline{\boldsymbol{a}}$ is obtained from (10). The variance of the noise $v[m,n]$ can be calculate using

$$\sigma_v^2 = (\underline{\boldsymbol{a}}^T\underline{\boldsymbol{a}})^{-1}\underline{\boldsymbol{a}}^T\underline{\boldsymbol{r}}_1 \tag{11}$$

The weighting diagonal matrix, $\underline{\boldsymbol{D}}$, is determined as (12).

$$D[i,i] = \begin{cases} 1 & \text{for } 1 \le i \le (p_1+1)(p_2+1) - 1 \\ \\ \lceil 50\sigma_v^2 \rceil & \text{for } (p_1+1)(p_2+1) \le i \le 2(p_1+1)(p_2+1) - 2 \end{cases} \tag{12}$$

where $\lceil x \rceil$ denotes rounding toward infinity.

6 Possible Neighbourhood Around Tumour

The main purpose of this paper is to find out whether the set of AR model coefficients that represents the texture of the block containing an tumour can be

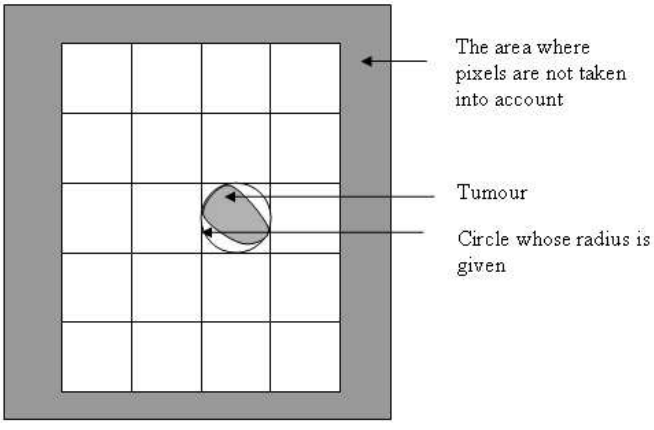


Fig. 1. The possible neighbourhood of the block containing a tumour in a mammogram

found anywhere else in the mammogram. For all the mammograms used in the simulations the information on the location and the size of the tumour is given. From this piece of information, we establish the possible neighbourhood by calculating the number of the blocks with the same size of the block containing the tumour we can have to four corners of the mammogram. In general, for the small tumours, we have a large number of blocks in the possible neighbourhood, whereas for the big tumours, we have small number of blocks in the possible neighbourhood. However, the total number of blocks in the possible neighbourhood also depends on the location of the tumour. All the blocks in the possible neighbourhood are restricted to have equal size, so some pixels near the borders might be neglected. An illustration can be found in Fig. 1. The smallest block in these simulations is 13×13 and the largest block is 395×395.

7 Simulation Results

From the database [5], the mammograms with tumours are extracted and their identity number remains as shown under (1) in Table 1. For each mammogram, the square block size that contains the tumour can be found under (2) in Table 1. The possible neighbourhood is defined and the number of blocks in the neighbourhood can be found under (3). A set of AR model coefficients is estimated using the method in Section 5. The order of the model is assumed to be 1×1. Each set of AR model coefficient is compared with the set representing the block with the tumour. The number of sets of AR model coefficients that exceed $\pm 3\%$ of the coefficients representing the tumour is counted for each mammogram and the results can be found under (4) in Table 1. The percentage of these sets in all blocks in the possible neighbourhood can be obtained under (5) in the same table.

For 100 tumours in 95 mammograms from the database, 38 of them contain no repeating AR model coefficients in their possible neighbourhood. For those

Table 1. The results arisen from the comparison of two-dimensional autoregressive model coefficients of the tumour block and its possible neighbourhood

(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
mdb001	395 × 395	1	0	0	**mdb144b**	55 × 55	323	4	1.24
mdb002	139 × 139	41	2	4.88	**mdb145**	99 × 99	99	0	0
mdb005a	61 × 61	255	1	0.39	**mdb148**	349 × 349	4	0	0
mdb005b	53 × 53	341	2	0.59	**mdb150**	125 × 125	48	2	4.17
mdb010	67 × 67	209	1	0.48	**mdb152**	99 × 99	99	2	2.02
mdb012	81 × 81	144	0	0	**mdb155**	191 × 191	19	0	0
mdb013	63 × 63	255	1	0.39	**mdb158**	177 × 177	24	0	0
mdb015	137 × 137	35	0	0	**mdb160**	123 × 123	48	1	2.08
mdb017	97 × 97	99	1	1.01	**mdb163**	101 × 101	89	4	4.49
mdb019	99 × 99	99	2	2.02	**mdb165**	85 × 85	120	2	1.67
mdb021	99 × 99	80	0	0	**mdb167**	71 × 71	168	0	0
mdb023	59 × 59	271	6	2.21	**mdb170**	165 × 165	24	0	0
mdb025	159 × 159	29	0	0	**mdb171**	125 × 125	55	5	9.09
mdb028	113 × 113	63	1	1.59	**mdb175**	67 × 67	209	4	1.91
mdb030	87 × 87	120	5	4.17	**mdb178**	141 × 141	41	0	0
mdb032	133 × 133	48	0	0	**mdb179**	135 × 135	41	4	2.88
mdb058	55 × 55	323	3	0.93	**mdb181**	109 × 109	71	0	0
mdb063	67 × 67	209	9	4.31	**mdb184**	229 × 229	16	0	0
mdb069	89 × 89	109	0	0	**mdb186**	99 × 99	100	0	0
mdb072	57 × 57	305	11	3.61	**mdb188**	123 × 123	48	1	2.08
mdb075	47 × 47	461	5	1.08	**mdb190**	63 × 63	239	0	0
mdb080	41 × 41	624	3	0.48	**mdb191**	83 × 83	120	0	0
mdb081	263 × 263	8	0	0	**mdb193**	265 × 265	8	0	0
mdb083	77 × 77	155	1	0.65	**mdb195**	53 × 53	341	5	1.47
mdb090	99 × 99	89	2	2.25	**mdb198**	187 × 187	15	1	6.67
mdb091	41 × 41	599	0	0	**mdb199**	63 × 63	239	2	0.84
mdb092	87 × 87	120	2	1.67	**mdb202**	75 × 75	155	3	1.94
mdb095	59 × 59	288	10	3.47	**mdb204**	43 × 43	528	5	0.95
mdb097	69 × 69	195	2	1.03	**mdb206**	35 × 35	840	6	0.71
mdb099	47 × 47	461	6	1.3	**mdb207**	39 × 39	675	4	0.59
mdb102	77 × 77	143	7	4.9	**mdb209**	175 × 175	24	0	0
mdb104	101 × 101	99	1	1.01	**mdb211**	27 × 27	1481	28	1.89
mdb105	197 × 197	19	0	0	**mdb213**	91 × 91	109	0	0
mdb107	223 × 223	15	1	6.67	**mdb214**	23 × 23	2069	1	0.0483
mdb110	103 × 103	80	3	3.75	**mdb218**	17 × 17	3968	2	0.0504
mdb115	235 × 235	9	0	0	**mdb219**	59 × 59	271	2	0.74
mdb117	169 × 169	24	0	0	**mdb222**	35 × 35	869	0	0
mdb120	159 × 159	35	0	0	**mdb223a**	59 × 59	271	0	0
mdb121	165 × 165	19	0	0	**mdb223b**	13 × 13	7055	18	0.2551
mdb124	67 × 67	209	4	1.91	**mdb227**	19 × 19	3135	63	2.01
mdb125	121 × 121	63	0	0	**mdb231**	89 × 89	120	3	2.5
mdb126	47 × 47	440	8	1.82	**mdb236**	29 × 29	1259	3	0.24
mdb127	97 × 97	89	1	1.12	**mdb238**	35 × 35	840	4	0.4762
mdb130	57 × 57	288	8	2.78	**mdb239a**	81 × 81	131	0	0
mdb132a	105 × 105	72	0	0	**mdb239b**	51 × 51	360	23	6.39
mdb132b	37 × 37	728	6	0.82	**mdb240**	47 × 47	440	45	10.23
mdb134	99 × 99	89	0	0	**mdb241**	77 × 77	168	2	1.19
mdb141	59 × 59	288	6	2.08	**mdb244**	105 × 105	63	2	3.17
mdb142	53 × 53	341	1	0.29	**mdb248**	21 × 21	2499	6	0.24
mdb144a	59 × 59	288	0	0	**mdb249**	97 × 97	89	0	0

(1) Mammogram identity number (as in the Database [7])
(2) The size of the block that contains a tumour
(3) The number of blocks available in the entire mammogram.
(4) The number of blocks that can be represented in 2-D AR model coefficients similar to the set representing the tumour block.
(5) The percentage of blocks that contain similar 2-D AR model coefficients compared to the set representing the tumour block.

who do have a set or sets of AR model coefficients within the range of ±3% of the AR model coefficients representing the tumour block, mammogram mdb240 contains 10.23% of the AR model coefficients in all 440 sets estimated which are within this range, followed by mammogram mdb171, which has 9.9% of its 55 blocks in the neighbourhood within this range of the AR model coefficients representing the tumour. For all the tumours in mammograms we took into account, 98.6% of the time, the AR model coefficients (within ±3%) representing the tumour block does not appear again in any other blocks in the possible neighbourhood.

8 Conclusion and Summary

In this work, we looked into the possibility of representing the tumour area of a mammogram using two-dimensional $(2-D)$ autoregressive (AR) models. The possible neighbourhood with blocks of the same size as the one containing a tumour is defined and a set of AR model coefficients is estimated for each block. For all the sets of coefficients obtained, we compared them with the set representing the tumour. It was found from the simulations results that 98.6% of the time, the AR model coefficients representing the tumour can not be found anywhere else in the same mammogram using the same block size. This lead to the conclusion that $2-D$ AR model coefficients can provide sufficient data for texture analysis of mammograms.

References

1. S.M. Kay, *Modern spectral estimation: theory and application*, Prentice Hall 1988.
2. S. Lee and T. Stathaki, "Texture Characterisation Using Constrained Optimisation Techniques with Application to Mammography", Fifth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2004), on CD-ROM.
3. S. Lee and T. Stathaki, "Two-Dimensional Autoregressive Modelling Using Joint Second and Third Order Statistics and a Weighting Scheme", Twelfth European Signal Processing Conference (EUSIPCO), on CD-ROM, 2004.
4. S. H.-H. Lee, *Novel Approaches to Two Dimensional Autoregressive Modelling*, Ph.D. and DIC thesis, Imperial College London, 2004.
5. The Mammographic Image Analysis Society (MIAS): MiniMammography Database. http://www.wiau.man.ac.uk/services/MIAS/MIASmini.html, last access: 11th February 2005.
6. T. Stathaki, "2-D autoregrssive modelling using joint and weighted second and third order statistics", *Electronics Letters*, **32**(14) (1996) 1271–1273.
7. J. Suckling, J. Parker, D. Dance, S. Astley, I. Hutt, C. Boggis, I. Ricketts, E. Stamatakis, N. Cerneaz, S. Kok, P. Taylor, D. Betal, and J. Savage, "The Mammographic Image Analysis Society Digital Mammogram Database", *Exerpta Medica*, International Congress Series, **1069** (1994) 375–378.
8. A. Swami, G.B. Giannakis and J.M. Mendel, "Linear modeling of multidimensional non-Gaussian processes using cumulants", *Multidimensional Systems and Signal Processing*, **1** (1990) 11–37.

Texture Analysis of CT Images for Vascular Segmentation: A Revised Run Length Approach

Barbara Podda[3] and Andrea Giachetti[1,2]

[1] Universitá di Cagliari, Dip. Matematica e Informatica,
via Ospedale 72, 09124 Cagliari
giach@unica.it
[2] CRS4 - POLARIS, Loc. Piscinamanna 09010 Pula (CA), Italy
[3] DIEE, Universitá di Cagliari, Piazza d'Armi, 09123, Cagliari

Abstract. In this paper we present a textural feature analysis applied to a medical image segmentation problem where other methods fail, i.e. the localization of thrombotic tissue in the aorta. This problem is extremely relevant because many clinical applications are being developed for the computer assisted, image driven planning of vascular intervention, but standard segmentation techniques based on edges or gray level thresholding are not able to differentiate thrombus from surrounding tissues like vena, pancreas having similar HU average and noisy patterns [3,4]. Our work consisted in a deep analysis of the texture segmentation approaches used for CT scans, and on experimental tests performed to find out textural features that better discriminate between thrombus and other tissues. Found that some Run Length codes perform well both in literature and experiments, we tried to understand the reason of their success suggesting a revision of this approach with feature selection and the use of specifically thresholded Run Lengths that improves the discriminative power of measures reducing the computational cost.

1 Introduction

Digital diagnostic imaging modalities are providing the medical community with a large amount of high quality data that can be useful to obtain a better understanding of different pathologies. Computer assisted analysis of these images is fundamental to recover in a fast and objective way models, parameters and morphological descriptor of human organs. One of the most promising application of morphological reconstruction from digital images is the creation of patient specific vascular models from contrasted CT scans. Through the use of contrast media and helical multi-slice CT scanners it is possible to recover 3D models with sub-millimeter accuracy applying segmentation techniques based on data thresholding at specific HU values or more complex deformable models exploiting the HU level constancy of the contrast medium. Similar results cannot be obtained for the external walls of the vessel. In fact, especially in the aorta, it is possible to have deposits of thrombotic tissue between the blood and the vessel wall. These regions are not enhanced by the contrast liquid and their HU values are close to those of other tissues that can be found near the vessel. A few

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 907–914, 2005.

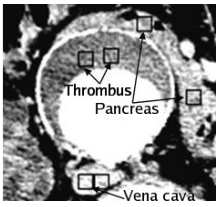


Fig. 1. Textures creating problems in aortic segmentation are thrombus, vena and pancreas, having close HU value averages

solutions to this problem have been proposed in literature. Olabarriaga et al. [3] trained a k-NN classifier to recognized vessel borders from samples taken on a line perpendicular to the vessel surface, Giachetti et al. [4] also analyzed 1D profiles giving ad hoc rules driving a deformable contour. Results are not always reliable, so that shape constraints have been also proposed ([1,4]) to avoid the inclusion in the vascular volume of voxels belonging to pancreas or vena cava. In this work we aimed at pointing out if this segmentation problem can be at least reduced by looking at texture descriptors. Two main questions have to be solved for this purpose: is there relevant information in the CT texture of the tissues of our interest? It is possible to find simple texture features that can be used in classical segmentation algorithms?

The paper is organized as follows: a short review of approaches for CT texture analysis is presented in Section 1; Section 2 presents an analysis of Thrombotic texture and a discussion about the meaning of the features based on Run Length Matrices, the most successful ones in this context. Section 3 presents experimental results supporting the analysis of the previous section and suggesting the possibility of designing ad hoc texture descriptors that can be used to improve thrombus segmentation.

2 Texture Analysis and CT: A Review

In recent years many studies based on texture analysis of CT images [5,9,8,12] have been proposed to help the process of tissues and organs identification for medical applications. These studies suggest that some features, especially those based on Grey Level Run Length Matrix (GLRLM) are able to discriminate between different tissues types. Mir et al.[5] analyzed textural features based on Grey-Level Co-Occurrence Matrix (GLCM), Grey Level Run Length Matrix (GLRLM), and Grey Level Difference Matrix (GLDM). They used as input for the study CT Images of the abdomen with depth reduced to 256 levels and identified some features able to recognize the onset of disease in liver tissue which could not be identified even by trained human operators. In [9] Run Length Matrices are successfully applied to assess the spinal fracture risk from CT texture. Koss et al. [11] applied Haralick's Co-occurrence Matrix based features to classify abdominal tissues (not including those of our interest). They classified

pixels on 6 texture classes using a neural network classifier, finding an accuracy of about 90%. More recently, Raicu et al.[12] presented a texture dictionary for anatomical structures. They extracted features based on co-occurrence and run length matrices from CT Images. The features analyzed provided useful information about five different abdominal organs that could not be appreciated by human eye. In particular, High Grey Run Emphasis and Sum Mean proved to be the most discriminative indexes for the process of organs identification. In [8] the definition of Run Length Matrix is extended to the three dimensional space, in order to evaluate more properly textural features based on GLRLM for volumetric image data such e.g CT scans.They put emphasis on the fact that using many gray levels intensities, many runs would contain only one pixel, and consequently it is necessary to reduce the number of levels. They quantized the original 16 bit CT images into 32 gray-levels using linear mapping. The High Grey Level Run Emphasis and the Long Grey Level Run Emphasis descriptors proved to have a strong similarity for 2D and volumetric texture data. The usual approach in papers dealing with texture analysis is to compute a lot of features and give them as an input to a classifier. To reduce the dimensionality of the feature space Singular Value Decomposition or a more meaningful Linear Discriminative Analysis are used.[9,11,8]. Our goal is slightly different, because we are not interested in building a classifier, but in computing one or a few features that could be inserted easily as constraints in a segmentation tool based, for example, in deformable surfaces or region growing. This is why we concentrated our efforts in feature selection and in understanding features' meaning for CT images of thrombus.

3 Texture Analysis of CT

Is CT texture meaningful? The problem of our interest is the discrimination of thrombus from surrounding tissues in abdominal CT. The main question is: can we distinguish between tissues that may be connected with the thrombotic region and the thrombus itself using texture information? A closer look at the histogram of thrombus (Fig. 2A) reveals that it is not Gaussian like that computed on a CT scan made with similar protocol on a phantom model of a material with constant density (Fig. 2B). Texture may therefore encode non trivial information about the tissue. This is not surprising, considering the literature results showing that Run Length Matrix based features are able to discriminate tissues on CT images. We are therefore interested in finding the reason why they are successful.

Run Length Matrices based features. RLM based features are computed in the following way: first the gray levels are subsampled in a coarser range of N values. Then runs of equal levels with lengths from 1 to M along a defined direction are computed. Runs are stored in a MxN matrix called Run Length Matrix and 11 values are usually derived from it: Short Run Emphasis (SRE), Long Run Emphasis (LRE), Gray Level Non Uniformity (GNU), Run Length Non Uniformity(RNU), Run Percentage (RPC), Low Gray Level Run Emphasis

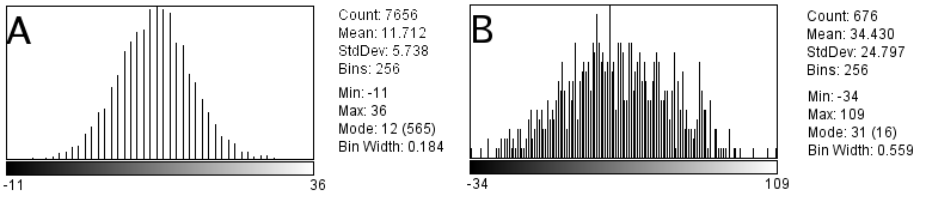


Fig. 2. While the histogram of the CT of a truly homogeneous region (left) is approximately Gaussian in detail, thrombus histogram (right) is, on a fine scale, complex, apparently coming from 2-3 overlapping distributions

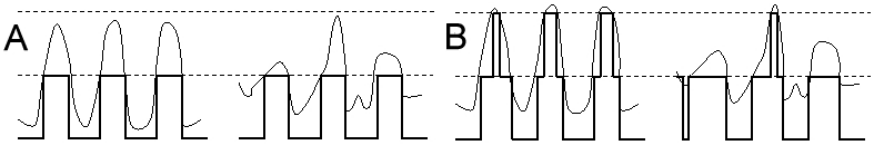


Fig. 3. Simple 1D example showing that just shifting a little the quantization limits, the gray level clustering used as a preprocessing before computing run length can miss (A) or capture (B) texture differences

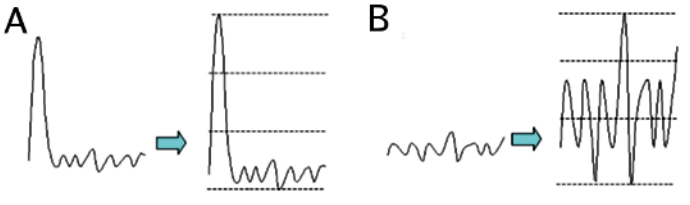


Fig. 4. If we reduce the original level range mapped into the reduced number of grays to the min-max interval, an outlier can differentiate two similar textures causing completely different runs (A,B)

(LGRE), High Gray Level Run Emphasis (HGRE), Short Run Low Gray Level Run Emphasis (SRLGE), Short Run High Gray Level Run Emphasis (SRHGE), Long Run Low Gray Level Run Emphasis (LRLGE), Long Run High Gray Level Run Emphasis (LRHGE) (see [7,10,8]). If texture is not directional (as in our case), measures are usually averaged on all directions. Our hypothesis for the success of some RLM features on CT is that the texture is due to "grains" of different materials and sizes, and run lengths measure size and orientations of grains. With this interpretation, RLM methods can be interpreted as follows: the gray level subsampling acts as a "material classification" and the run length matrices measure statistical features of the classified regions.

The role of depth reduction. The critical step in the tissue discrimination is probably the pre-processing, i.e. the gray level subsampling performing the rough classification of the materials composing the image texels. This step is not,

however, discussed in papers describing applications of the method. Without a gray level reduction, RLM is clearly not meaningful, being most of the runs equal to 1. The matrix becomes zero everywhere except for the first column, containing the image histogram. In this limit, the few RLM-dependent features that appear meaningful in experiments converge to quadratic means, that can have discriminative power, but do not add new information to simple histogram based features. Clustering grays in a reduced number of levels, changes this result, making the discrimination obtained with some of the RLM features less correlated with histogram based measurements but still useful (see experimental section). Another fact to be considered is that the clustering function should not be a simple linear mapping of the full range or the min-max range of the region into the selected number of levels. In the first case, especially if we want to use a small number of gray levels, the clustering may miss meaningful structures, while in the second the effect of noise is too relevant and a single outlier pixel can change completely the transformed map, the corresponding homogeneous regions, the run lengths and the output of the texture descriptors. To understand these facts, let us consider the following situation: a 1D image, described by a continuous function (Fig.3). Run lengths computed at a coarse quantization, correspond to the length of the constant lines of the quantized functions. The coarse quantization in A, creates the same runs from two completely different signals, and no features extracted from them could be used to discriminate the two texture. Different runs can be obtained not only with a finer quantization, but also with the same number of grays just taking quantization limits slightly shifted (B). Fig. 4 shows that for our application, a min-max rescaling of the gray levels before the quantization, performed in some applications before the gray level resampling, is not recommended: consider the 1D textures in A,B; they can be the same kind of tissue with just an outlier, or a small portion of another material on the left. The min-max transform creates two completely different signals that creates with the quantization limits of the dashed lines completely different runs. These considerations suggested us a new approach to compute gray level Run Length Matrices: knowing the HU value range and variations of the interested tissues we can create the low-depth image used for the feature computation clustering gray levels in a custom way, putting ad hoc thresholds, equal for all the ambiguous tissues. In this way it is possible to compute features from a very small number of levels (i.e. 4) without losing discriminative power. We therefore designed experiments aimed at selecting the ideal number of gray levels and the ideal thresholding function to extract the most powerful and efficient RLM features.

4 Experimental Results

To verify other authors' results, a comparative analysis of the textures was performed on classified image regions taken from 3 CT datasets of over 500 images acquired with a spiral CT scanner at resolution of about 0.6x0.6x2mm. We considered the three textures generating problems to usual segmentation algorithms

Table 1. Intra and inter class Texture Features distances measured on Thrombus, Vena Cava and Pancreas from the original 12 bit images. Features considered are gray level mean, Co-Occurrence Matrix based (CM), Gray Level Difference Matrix based (CM) and Gray Level Run Length Matrix based (RLM).

Distance	mean	CM1	CM2	CM3	CM4	CM5
T-T	0.24	0.85	0.85	0.06	1.06	0.13
T-P	2.77	0.33	0.31	0.65	2.01	0.08
T-V	5.89	0.11	0.11	1.89	0.59	0.25

Distance	DM1	DM2	DM3	DM4	DM5
T-T	0.13	0.20	0.08	0.10	0.51
T-P	0.08	0.75	0.35	0.44	0.91
T-V	0.25	1.67	1.42	1.44	1.65

D.	RLM1 SRE	RLM2 LRE	RLM3 GLNU	RLM4 RLN	RLM5 RPC	RLM6 LGRE	RLM7 HGRE	RLM8 SRLGE	RLM9 SRHGE	RLM10 LRHGE	RLM11 LRHGE
TT	0.25	0.32	0.19	0.20	0.20	0.24	0.23	0.15	0.27	0.39	0.08
TP	0.21	0.19	1.36	0.23	0.21	2.74	2.86	2.86	2.47	1.49	1.81
TV	0.05	0.05	1.39	0.05	0.00	5.79	5.93	5.75	5.27	2.74	3.45

for the definition of thrombotic region limits, i.e. Thrombus itself, Vena Cava and Pancreas. The goal of the first series of experiments this was to find if there are measures different from the gray level average, that are sufficiently constant inside a class and different from the other classes. We considered the main texture features used in the papers reviewed in Section 1, i.e. features based on first order statistics, on Gray Level Co-Occurrence Matrices, Gray Level Difference Matrices and Run Length Matrices and tested all of them on the selected classified windows. We evaluated the discriminative power of features by measuring them on squared 10x10 windows correctly labeled as "Thrombus", "vena cava" and "pancreas". We computed texture features on these windows and their intra and inter-class averages and standard deviations. In detail, we computed on training sets of windows intra-class distance inside thrombus (T-T) and the inter-class distance between thrombus and pancreas (T-P) and thrombus and vena cava (T-V). The distance has been computed as follows: $D_f(X_1, X_2) = |F(X_1) - F(X_2)|/\sigma$ where X_1 and X_2 are two squares of 10x10 pixels; $F(X_1)$ and $F(X_2)$ are the features F computed into X_1 and X_2; σ is the average standard deviation of the two classes. Table 1 shows the results obtained, showing clearly that gray level average and several measures derived by the Run Length Matrix (also Gray level Co-Occurrence Matrix and Gray Level Difference Matrix), are able to discriminate between textures of our interest in CT images.

But what about feature correlation and complexity? Complexity of GLRLM, GLCM and GLDM is very high if we work with 12 bit images (matrices with 4096 rows).In many classification approaches this problem is not analyzed because the goal is offline classification and the idea is, like in [10], to compute a lot of features and then combine them in a small and more significant set using Linear Discriminant Analysis or PCA. Our goal is, however, different, so we are interested in selecting one or few features simple, discriminating and not correlated with the average in order to improve the segmentation based on gray level. Tang ([10]), demonstrated also a strong correlation of classic RLM features on Broadtz textures. We already proved that if depth is relevant, runs tends to 1 and features are also strongly correlated with the average gray level. This is

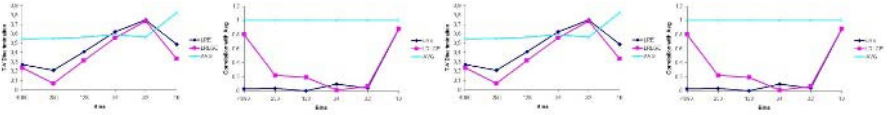


Fig. 5. Changes in features discriminative power and correlation with the (depth dependent) gray average due to bins subsampling. Features uncorrelated with average, like RLE still are uncorrelated and increase discriminative power up to 32 levels, then features lose their meaning. Using ad hoc thresholding, 8 levels are enough to preserve features discriminative power

Fig. 6. Example of snake segmentation exploiting an image force computed in a 2D feature space. Left: original image with superimposed manually drawn thrombus contour. Center: result of snake based segmentation with image force based on gray level thresholding. Right: results of enhanced segmentation exploiting also a RLE feature (HGRE) map to constrain the snake.

confirmed by experimental data: RLM features showing discriminative power in Table 1 appear strongly correlated with the mean (coefficient close to 1). They do not seem to add relevant information (discrimination values are indeed quite similar). However, linearly decreasing the depth, we observed that gray-dependent features like LRLGE, SRHGE lose correlation with the mean and increase their discriminative power (see Fig 5, from data of the noisiest dataset available) and features, like LRE, having small discriminative power at original resolution, becomes clearly discriminative, maintaining their low correlation with the gray level average.

This behavior is evident up to 32 levels, with less performances are worse or correlation higher, probably because the correct thresholds for the texels differentiation are lost. But, even with just 8 levels, if threshold are selected ad hoc as suggested in the previous sections around the thrombus HU average, discriminative power of measures like LRE is preserved. This fact demonstrates that exist (a few) features very fast to be evaluated that are uncorrelated with the gray level average and add relevant discriminative power.

Many other tests will be necessary to better understand the features' behavior, it is however clear that texture features may be useful for thrombus delimitation and could be introduces also in fast vascular segmentation tool. As an example, let us use a feature to constrain deformable models. The idea is simple: image forces stopping balloon models inflation are computed in a feature space and do not depend only by local gray level values. An example of preliminary result of texture enhanced contour segmentation is shown in Fig. 5.

The application of texture filters in semi-automatic segmentation tools is however quite difficult because of problems in finding correct thresholds automatically and the low resolution and noise of texture descriptors maps.

5 Discussion

Texture analysis can improve the quality of the segmentation of ambiguous regions in CT images. Simple and effective features based on Run Length Matrices can be used in vascular segmentation when thrombotic tissue is not easily differentiated by surrounding tissues like vena cava. Our analysis show that the image depth reduction step, usually performed without an adequate discussion, has a great importance in determining the discriminative power of RLM based features. New tests will be performed on larger datasets to investigate feature behavior (also GLCM based) and the use of feature-space image forces in 2D and 3D deformable models will be investigated and tested.

References

1. M. de Bruijne et al, "Active shape model based segmentation of abdominal aortic aneurysms in CTA images". Proc. SPIE Medical Imaging 2002:Image Proc. 4684, 463–474 (2002)
2. M. Subasic, S. Loncaric, E. Sorantin, "3-D image analysis of abdominal aortic aneurysm", Proc. SPIE Medical Imaging vol. 4684, pp. 1681-1689, 2002
3. S.D. Olabarriaga, M. Breeuwer, W.J. Niessen, "Segmentation of Abdominal Aortic Aneurysms with a Non-parametric Appearance Model", LNCS 3117, ECCV Workshops CVAMIA and MMBIA, pp.257-268 (2004).
4. A.Giachetti and G.Zanetti, "AQUATICS Reconstruction Software: The Design of a Diagnostic Tool Based on Computer Vision Algorithms". LNCS 3117, ECCV Workshops CVAMIA and MMBIA, pp. 48-63 (2004).
5. A.H. Mir, M. Hanmandlu, S.N. Tandon, "Texture Analysis of CT Images", IEEE Engineering in Medicine and Biology, 14:6 pp. 781-786 (1995)
6. R.M. Haralick, K. Shanmugam, Its'Hak Dinstein, "Textural Features for Image Classification", IEEE Trans. on Systems, Man, and Cybernetics, 3,6:610-621(1973)
7. M.M. Galloway, "Texture Classification using grey level run lengths", Comp. Graphics and Image Processing, 4: 172-179 (1975)
8. D.H. Xu et al., "Run Lenght Encoding for Volumetric Texture", Proc. of the 4th IASTED Int. Conf. on Visualization, Imaging, and Image Processing, 2004.
9. M. Ito et al., "Trabecular texture analysis of CT images in the relationship with spinal fracture." Radiology. 1995 Jan;194(1):55-9
10. X, Tang, "Texture information in run-length matrices" IEEE Trans. on Image Processing, **7**:11 , pp. 1602–1609 (1998)
11. J. E. Koss et al., "Abdominal Organ Segmentation Using Texture Transforms and a Hopfield Neural Network", IEEE Trans.on Medical Imaging, **18**:7 (1999).
12. D. S. Raicu et al., "A Texture Dictionary for Human Organs Tissues' Classification", Proc. 8th World Multiconf. on Syst., Cyb. and Informatics, Orlando, USA (2004)

The Spiral Method Applied to the Study of the Microcalcifications in Mammograms

Sergio Vitulano, Andrea Casanova, and Valentina Savona

Dipartimento di Scienze Mediche Internistiche,
Università di Cagliari, via San Giorgio 12, 09124 Cagliari, Italy
vitulano@pacs.unica.it

Abstract. In this paper a linear transformation, the spiral method, is introduced; this transformation maps an image represented in a 3-D space into a signal in a 2-D space. Some features of the spiral are presented: for instance the topologic information of the objects in the image, their contours, areas and the shape of the objects themselves. Two different case-study are presented: the use of spiral method in order to evaluate the number, the size, the shape and the location of the microcalcifications by the use of signals related to the mammograms; entropy is proposed as a measure of the degree of the parenchyma disorder of the mammograms and its use for a system CAD.

1 Introduction

The precocious diagnosis of the lesion in the breast has an absolutely determining role in the evolution and in the final result of the lesion.

As a matter of fact, the recognition of a precocious malignant lesion has a favourable result in 90% of the cases, while it is lethal in 50% of the cases, when a secondary metastasis is induced.

Therefore, the mass of screening, which is applied to the mammography, gets a very important role: 30-50% of the precocious lesions are made of agglomerates of micro calcifications, which dimensions are between 0,5 -1 mm [1].

Only through an accurate and skilled analysis of mammograms, it is possible to point out isolated micro calcifications. Indeed, there is a strong variability among their shapes, dimensions, and besides, they are immersed in a parenchyma tissue, altered in its structure by the existing lesion.

The formulation of a diagnosis, only through the analysis of mammograms, is definitely very complex, especially if compared to those ones that use CT or MRN. It is also the origin of 30% of mistakes done by the radiologists [2] [3]. The proper methodologies of the image analysis and of the Pattern Recognition have been used by many CAD systems (Computer Aided Diagnosis). The CAD systems have the purpose to help the radiologist in the radiogram analysis. They want to be a second opinion, not a substitute of the same radiologist. [4] [5].

In a previous work [8], we have introduced a CAD that applied entropy as a measure of parenchyma disorder, represented in a mammogram, in order to point out the differences between the benignant and malignant lesions for the masses and for the micro calcification.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 915 – 921, 2005.

The concentration of microcalcifications, their shape and dimension are useful information to evaluate correctly the entity of the lesion.

In this work, we wish to supply with a measure of the number of the existing micro calcifications in the area indicated by the radiologist; the contour and the area of each micro calcification located and their shape.

The paper is organized as follows: the first chapter describes the spiral method and some of its properties; the second chapter shows the results obtained by applying the proposed method to images containing benignant or malignant micro calcifications, and finally, the work will provide a discussion and authors' conclusions.

2 The Spiral Method

There are different methods meant to read the information contained in a digital image: in rows, in columns, or by recurring to paths such as the Peano's one. The choice of the scansion method is connected to the type of information that somebody wants to pick out from the image (e.g. a certain recurrence in a direction, the search of the points of maximum or minimum of the surface image in order to carry out the histogram, the time of the calculus etc).

We propose the spiral method for the scansion of a digital image in order to perform expected target (connected pixels, set of pixels that locate the regions of the image etc.).

We define: $A_{m,n}$ as the domain of the surface image where (m, n) are respectively the number of rows and columns of A.

Only out of simplicity of expression, we place m=n, i.e. A is a square matrix.

Definition 1. We define crown of the matrix C_1 the set of the pixels

$$C_1=\{ a_{1,1}\ldots a_{1,n};\ a_{2,n}\ldots a_{m,n};\ a_{m,n-1}\ \ldots . a_{m,1}\ ;\ a_{m-1,1}\ldots . a_{2,1}\} \tag{1}$$

that is, the order set of pixels contained in the row m=1 of the matrix, in the n-th column, in the m-th row, in the first column except the pixel, $a_{1,1}$ since it is already contained in the first row.

Let P is a discrete mono dimensional signal, so that:

$$P(x) = Px_1,\ Px_2,\ldots..,\ Px_i,\ldots Px_k \tag{2}$$

Definition 2.
Therefore, we define first differential

$$\Delta^1 Px_i = Px_{(i+1)} - Px_i \tag{3}$$

Definition 3.
We define second differential:

$$\Delta^2 Px_i = \Delta^1 Px_i - \Delta^1 Px_{(i-1)} \tag{4}$$

If we substitute the values $\Delta^1 Px_i - \Delta^1 Px_{(i-1)}$

$$\Delta^2 Px_i = \Delta^1 Px_i - \Delta^1 Px_{(i-1)} = Px_{(i+1)} - Px_i - (Px_i - Px_{(i-1)}) = \\ = Px_{(i+1)} - 2Px_i + Px_{(i-1)} \tag{5}$$

It is easy to verify that for every pixel belonging to a crown of the matrix, the second differential assumes value 0.

It is straightforward that if for three pixels, belonging to a crown, the relation (4) assumes value 0, then they are 4-connected with respect to the central pixel.

If we suppose $\boldsymbol{A_{m,n}}$ a bidimensional signal and $\boldsymbol{C_1, \ldots\ldots, C_k}$ the crown contained in its dominion, we define joined spiral to the signal $\boldsymbol{A_{m,n}}$ the relation:

$$T = U_{i=1,k}\, C_i \tag{6}$$

where $\boldsymbol{C_i}$ is the i-th crown obtained from the matrix $\boldsymbol{A_{m,n}}$.

It is important to observe that the relation (6) realizes a linear reversible transformation of a generic signal in a space 2-D in a signal in a space 1-D.

Therefore it follows:

$$\boldsymbol{A_{m,n}} \underset{\boldsymbol{S^{-1}}}{\overset{\boldsymbol{S}}{\Longleftrightarrow}} \boldsymbol{T_{mxn}} \tag{7}$$

Due to (7) a one to one application is established between each of the elements t_k belonging to T with each of the pixel $a_{i,j}$ of the matrix $A_{m,n}$.

The transformation S maintains the information regarding the form and the dimensions of the image domain, the topological information such as the number of the objects and their position, the area and the outline of the objects, etc.

For example, we assume t_k as the element to which corresponds the pixel $a_{i,j,}$ so the pixels 4-connected to $a_{i,j,}$ correspond to the elements in T for which the condition (8) is satisfied

$$\Delta^2 t_n = 0 \\ \text{or} \\ \Delta^2 t_n = 8 \tag{8}$$

The pixels of object in A are 4-connected, so :

1. The area of the object is given by the set V, whose elements satisfy the relation (8);
2. the contour of the object is given by the subset $V^1 \in V$ whose elements contain almost a ground pixel among its 8-connected pixels;
3. the signal T of the Fig. (2-c) supplies with a certain number of information about the shape of the set object V;
4. Concerning the elements belonging to the set V^1, we know:

a. **topological information** – from the abscissa of a point belonging to $\boldsymbol{T}$, we obtain the indexes of rows and columns of the pixels related to $\boldsymbol{A}$

b. **the shape of the object** – from the elements related to $V^1 \in V$, we are able to describe the shape of the object contained in $\boldsymbol{A}$

c. **shape 3-D of the object** – for each of the elements related to $\boldsymbol{V}$, we compute: its location in the domain of A (index of row and column) and its grey level. So it is possible to have both the information over the 3-D shape and to reconstruct $\boldsymbol{V}$ pixel by pixel.

In a previous work [9], we have proposed the HER (Hierarchical Entropy based Representation) method, as the algorithm meant to realize the information retrieval from a multidimensional database.

Briefly the relevant point about HER that a 1-D signal, T, may be represented by a string F, such: that:

$$T \approx F = m_1 e_1 ;\ m_2 e_2 ; \ldots\ldots\ldots ;\ m_k e_k \tag{9}$$

where

$\{m\} = m_{1,\ldots\ldots,} m_k$ are the maxima, extract in a hierarchical way from T;

$\{e\} = e_{1,\ldots\ldots,} e_k$ are the energies associated maxima $m_i \in \{m\}$

Let's suppose a 1-D signal *T*, where *m* and *M* corresponding to the minimum and maximum absolutes of *T* and E_T its total energy .

We define signal crest *C* the portion of the signal *T* between *m* and *M*.

In other words, the signal crest is obtained by placing the zero of the axes of ordinates equal to *m*.

We assume E_C the energy of the crest signal *C*.

We apply to the signal *C* the method HER , obtaining $\{m_k\}$ and $\{ e_k\}$ as the maxima and the energies of *C* respectively.

Let $e_i \in \{e_k\}$ the energy associated to the maximum of $m_i \in \{m_k\}$.

Definition 4.

We define entropy of the signal T the relation:

$$S_T = \frac{\sum_{i=1}^{K} e_i}{E_T} \tag{10}$$

It is important to underline that *m* and *M* aren't either the smaller or the bigger of the ordinate of the points of *T,* but the minimum and the maximum of the signal *T* (in a mathematical sense). It is straightforward that both the entropy of a constant signal (constant value of the function) and of a monotone signal (constant derivative) is equal to zero.

On the other hand the entropy equal to 1 corresponds to the maximum degree of disorder, i.e. there are not two points (x_i, x_{i+1}) with $i \in \{domainofSignal\}$ that have the same ordinate. In a previous work [10], we have introduced entropy as a measure of the disorder of a signal.

In the next section will show an application of our method to the analysis of mammograms.

3 Experimental Results

In a previous work [10], we have proposed a CAD system that employs the entropy of the parenchyma to discriminate between malignant and benignant lesions, both for masses and calcifications.

The mammograms taken into consideration belong to the DDSM (Digital Database for Screening Mammography, South Florida University) database, a collection of about 10.000 mammograms, digitized at 12 bits in a matrix of 2000x4000 pixels.

We have also verified that the recourse to the entropy to get retrieval information of a mammograms' database doesn't supply with valid results [8].

Two main problems arise from the study of microcalcifications: the number of the lesions and their shape in the selected area.

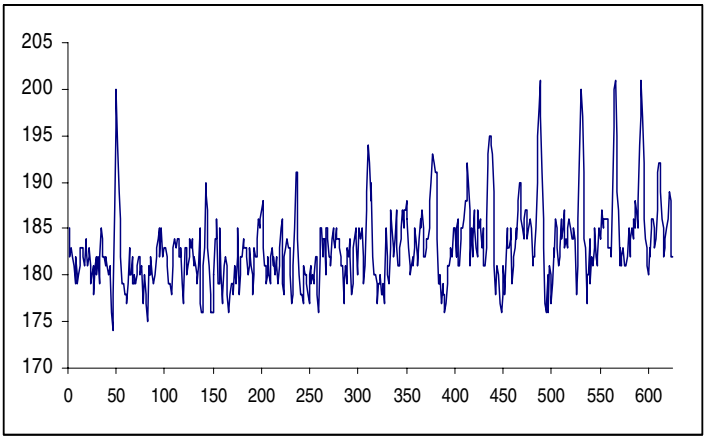


Fig. 1. The signal obtained by applying the spiral method to the portion of the mammogram shown in Fig.2b

We believe that the simple visual analysis, even performed from an expert radiologist, doesn't allow to obtain such information.

We strongly are persuaded that the linear transformation (7) allows a better handling of the information contained in the portion (subset) of the mammogram under exam.

In order to test our method, we have submitted 25 students from the medicine faculty, specialist in radiology, to a visual analysis of a certain number of mammograms containing both benignant and malignant microcalcifications. Each of the students were asked to describe each of the photograms; then we have shown them the equivalent signals obtained by applying the relation (7) and the results due to the analysis of their signals. The comparison of the obtained results showed that the analysis of the signals has allowed to get a more correct descriptions with respect to the ones due to the simple visual analysis of the images.

Let's describe the most relevant phases of the proposed method:

The physician selects from the digital mammogram the Region of Interest (even the whole mammogram), and visualizes this portion in the video device; then a grid is

overlapped, whose mesh dimension is an input parameter chosen by the physician. The grid is an useful help when a more detailed analysis has to be performed. Two sample portions of different mammograms, containing malignant and benignant micro calcifications, are shown in Fig.2 (a, b), while in Fig.3 (a, b) are shown the corresponding 1D signals obtained by using a mesh for the grid 25x25 pixels.

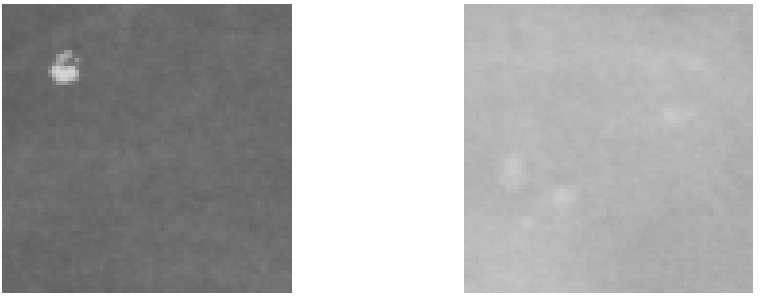
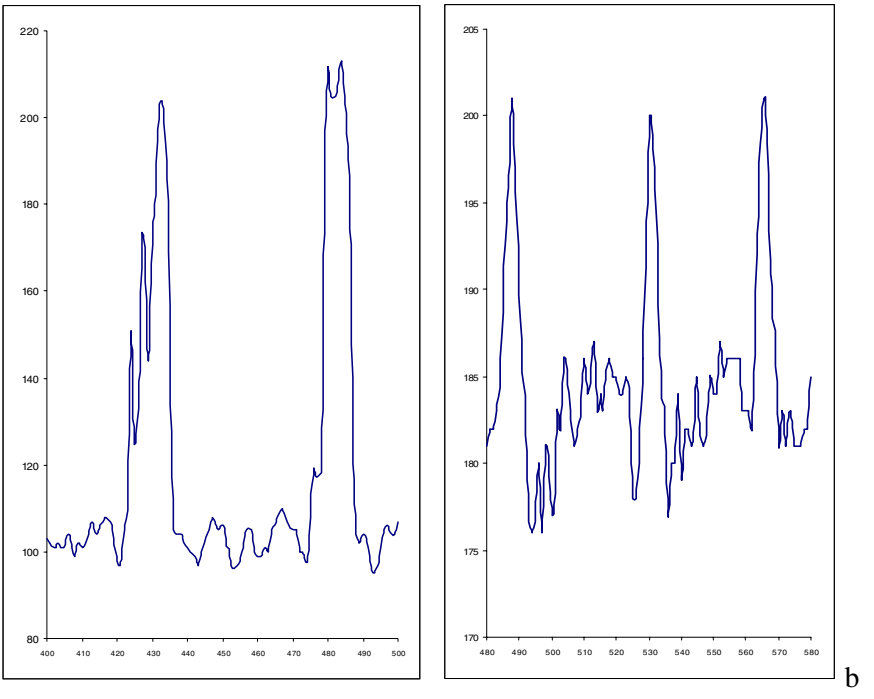


Fig. 2. A portion of mammogram with a benignant (a) and malignant microcalcifications (b) are shown



a b

Fig. 3. A portion of the signals related to Fig.1a and Fig.1b are shown respectively

The analysis and the diagnosis carried out from the study of the signals above mentioned appeared to be closest to the effective diagnosis.

We wish to emphasize the role of the number and of the shape of the maxima in order to classify the kind of the lesion. It is also interesting to compare the two signals in Fig 3 (a,b) and especially to go over the behaviour of the relating parenchyma, the grey level on the portion of the signals connected with the parenchyma in the considered area.

In conclusion we believe that the spiral method allows us to determine, in very simple way, the number of microcalcifications existing in the area of interest, their position, dimension and contour.

The spiral method, when compared to other methods proposed in the scientific literature, gives us the possibility to join a picture with an exact measure, to show the behaviour of the parenchyma and the shape of microcalcification with an algorithm that works in real time.

We think that the shape of the impulses of a signal is not enough to discriminate between benignant and malignant mammograms. Entropy as a measure of the parenchyma disorder has been successfully used in order to realize a CAD system with encouraging results. [6] [7] [11].

References

[1] R.N.Strickland "Wavelet transforms for dectecting microcalcification in mammograms" IEEE Trans. On medical imaging, Vol. 15 n.2 04/1996

[2] Qian Zhao et alt. "Multi-resolution source coding using Entropy constrained Dithered Scalar Quantization" Prooc. Of DCC'04 IEEE 2004

[3] M.Toews, T. Arbel "Entropy of likelihood feature selection for image correspondence. Prooc. ICCV'03 IEEE 2003

[4] M.Heath, et alt. (1998): 'Current status of digital database for screening mammography', (Kluwer Academic Pub.), pp. 457-460

[5] M.Melloul, L.Joskowicz (2002): 'Segmentation of microcalcification in X-ray mammograms using entropy thresholding', CARS 2002, H.U.Lemke et al. Editors.

[6] J.D.Ullman (1989) 'Principle of database and knowledge-based system', (Computer Science Press.).

[7] E.G.M.Petrakis, et alt. (1997): 'Similarity searching in medical image databases', IEEE Trans. Knowledge and Data Eng., 9, pp. 435-447

[8] A.Casanova, and S.Vitulano "Entropy As a Feature In The Analysis And Classification Of Signals"– Series on Software Engineering and Knowledge Engineering, N.15 - World Scientific ISBN 981-256-137-4

[9] A.Casanova, M.Fraschini "HER: Application on Information Retrieval" Series on Software Engineering and Knowledge Engineering Vol. 15 June 2003: 150-159 ISBN:981-238-587-8

[10] A.Casanova, V.Savona and S.Vitulano "The Role Of Entropy In Signal Analysis And Classification:An Application To Breast Diagnosis" Medicon2004 Ischia 2004

[11] A.Casanova, V. Di Gesù, G.Lo Bosco, S.Vitulano "Entropy measures in image classification" 4rd International Workshop Hmp04: Human And Machine Perception: (Santa Caterina di Pittinuri) Italy, September 2004. In press on Series on Software Engineering and Knowledge Engineering - World Scientific

Frequency Determined Homomorphic Unsharp Masking Algorithm on Knee MR Images

Edoardo Ardizzone, Roberto Pirrone, and Orazio Gambino

Universitá degli Studi di Palermo,
Computer Science and Artificial Intelligence Laboratory,
viale delle Scienze, Building 6, 3rd floor, Palermo, Italy

Abstract. A very important artifact corrupting Magnetic Resonance (MR) Images is the RF inhomogeneity, also called Bias artifact. The visual effect produced by this kind of artifact is an illumination variation which afflicts this kind of medical images. In literature a lot of works oriented to the suppression of this artifact can be found. The approaches based on homomorphic filtering offer an easy way to perform bias correction but none of them can automatically determine the cut-off frequency. In this work we present a measure based on information theory in order to find the frequency mentioned above and this technique is applied to MR images of the knee which are hardly bias corrupted.

1 Introduction

The RF inhomogeneity, also called Bias artifact, has been studied, in the latest years, in a lot of works related to this kind of artifact which afflict medical images and in particular Magnetic Resonance Images (MRI). The magnetic field of the device has intensity variations that depend on the scan sequence, the patient tissues and the kind of coil used. In particular, surface coil image is more sensible to this artifact but noise is not strong; on the contrary, body coil image has the opposite behavior. As a consequence of this phenomenon, the image is corrupted by a very low frequency signal whose visual effect is a non uniform illumination which can be seen either as white stripe, or as a more or less extended peak of luminance or an image shading along a direction where the magnetic field force lines go away. This is especially true if those images are produced by open coil devices like the one used for lower and upper limbs imaging. The approaches used to suppress this artifact can be subdivided in two classes: discovering the degradation model or modifying classic segmentation algorithms to classify the tissues taking into account the degradation model. In the first class of methods, a phantom,usually a container holding a known substance, may be introduced in the device with the aim to obtain an image afflicted by magnetic field intensity variations without any interaction with tissues and discover the degradation model. With this special experimental setup, each slice is corrected thanks to this special image that can be considered like a surface which approximates the illumination variations[7][10]. If body-coil image and surface-coil image of the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 922–929, 2005.

same slice are available, a preconditioned gradient algorithm can be used to discover the surface[2] after the input images have been registered. In [1] the bias artifact is modelled by Legendre polynomials and the algorithm estimates each uncorrupted class computing the image by a least-square technique, and assuming that it is made by piecewise zones with constant intensity level. The above methods can be considered evolutions of homomorphic filtering approaches[16] moving from[8] where the HUM (Homomorphic Unsharp Masking) has been developed. This approach either uses directly the phantom pixel to perform the correction or performing a log subtraction of the image with a blurred version of the same one using low pass filters[11]. In the last years has been developed the second class of methods oriented to modify classical segmentation algorithms to take into account, in their iterative steps, the artifact effect. This first approaches used the EM algorithm[12][6][4] while some more recent works deal with modified fuzzy c-means (fcm) algorithm[3][14][15][9]are used for their higher speed with respect to the previous one. An approach exclusively based on information theory is proposed in [13] which provides an unusual solution to the problem, minimizing the entropy of the inverse degradation model with Brent-Powell algorithm. The question is open because the NEUROVIA PROJECT [17] shows that this artifact isn't uncorrelated with tissues. This project was born as comparative study of six different algorithms with the aim to determine, thanks to a large and accurate experimental setup in various environment conditions, the best approach on encephalic MR: the result was that each algorithm shows a different bias estimation, due to its theoretical approach, and by the interaction of the tissues with the magnetic field. We perform our experimental setup on images of knee which are not shown in any study presented above and are corrupted by heavy bias artifact. The rest of the paper is arranged as follow. Sections 2 and 3 describe the proposed method for bias removal. In section 4 the experimental setup is detailed, while section 5 reports some conclusions.

2 The Adopted Filtering Scheme

Often in medical imaging the image is composed by a background and a foreground so if an homomorphic filtering is performed a streak artifact is produced on the boundary. Guillemaud [5] proposed an homomorphic filtering where the Region of Interest (ROI) is considered to avoid the artifact cited above. The algorithm is explained into detail as follow:

1. define a binary image (background=0;foreground=1) in order to define the region of interest named ROI
2. perform a log-transform to the original image with background suppressed:
$$I_{log} = log(I(ROI))$$
3. apply a low-pass butterworth filter of 2nd order in frequency domain obtained from the previous step:
$$I_{filt} = FFT^{-1}(Butterworth_{order=2}(FFT(I_{log})))$$

4. perform the same butterworth filtering to the binary image:
$$ROI_{filt} = FFT^{-1}(Butterworth_{order=2}(FFT(ROI)))$$
5. due to anti-transform operation, the images will be complex data, so perform magnitude of each pixel value:
$$I_{mag} = ABS(I_{filt}) \qquad ROI_{mag} = ABS(ROI_{filt})$$
6. divide pixel-by-pixel the ROI pixels of the image obtained at the step 3 with that one at step 4:
$$Log(Bias) = I_{mag}/ROI_{mag}$$
7. subtract the image at step 2 with that one obtained at the previous step and perform an exponential transform:
$$I_{restored} = exp(I_{log} - Log(Bias))$$
8. perform a contrast stretching of the image at step 7 in order to obtain the initial dynamic:
$$I_{corrected} = \frac{I_{restored} - \min(I_{restored})}{\max(I_{restored}) - \min(I_{restored})} \cdot \max(I) \tag{1}$$

The *Bias* image can be obtained in this way:

$$I_{Bias} = exp(Log(Bias)) \tag{2}$$

and performing the same stretching adopted in (1). An unresolved problem of this method is that the butterworth filter cutoff frequency (bfcf) is not specified: this is an important parameter because the entity of homomorphic process depends by this one. If it is too low, no effect will be visible on $I_{corrected}$; on the contrary, higher values of bfcf will produce on $I_{corrected}$ a loss of tissues contrast, that is tends to uniform gray level. For values of bcfc close to 1, produce a total inversion between $I_{corrected}$ and I_{Bias}: I_{Bias} will appear identical to the initial image I while $I_{corrected}$ is made by a quasi-uniform gray level in the ROI, as shown in fig.1.

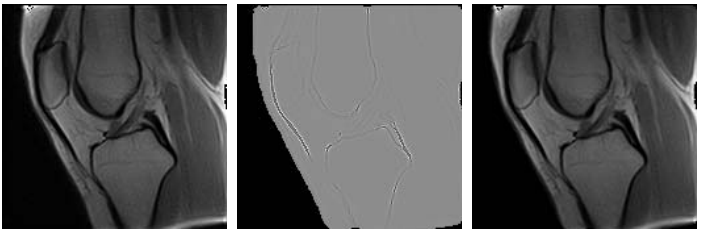


Fig. 1. Cut-off frequency 0.99 - From Left to Right: original, filtered and bias images

This fact is a classic image processing finding because step 4 corresponds to an high-pass filtering so the lower frequencies which are subtracted from I are given to I_{Bias}.

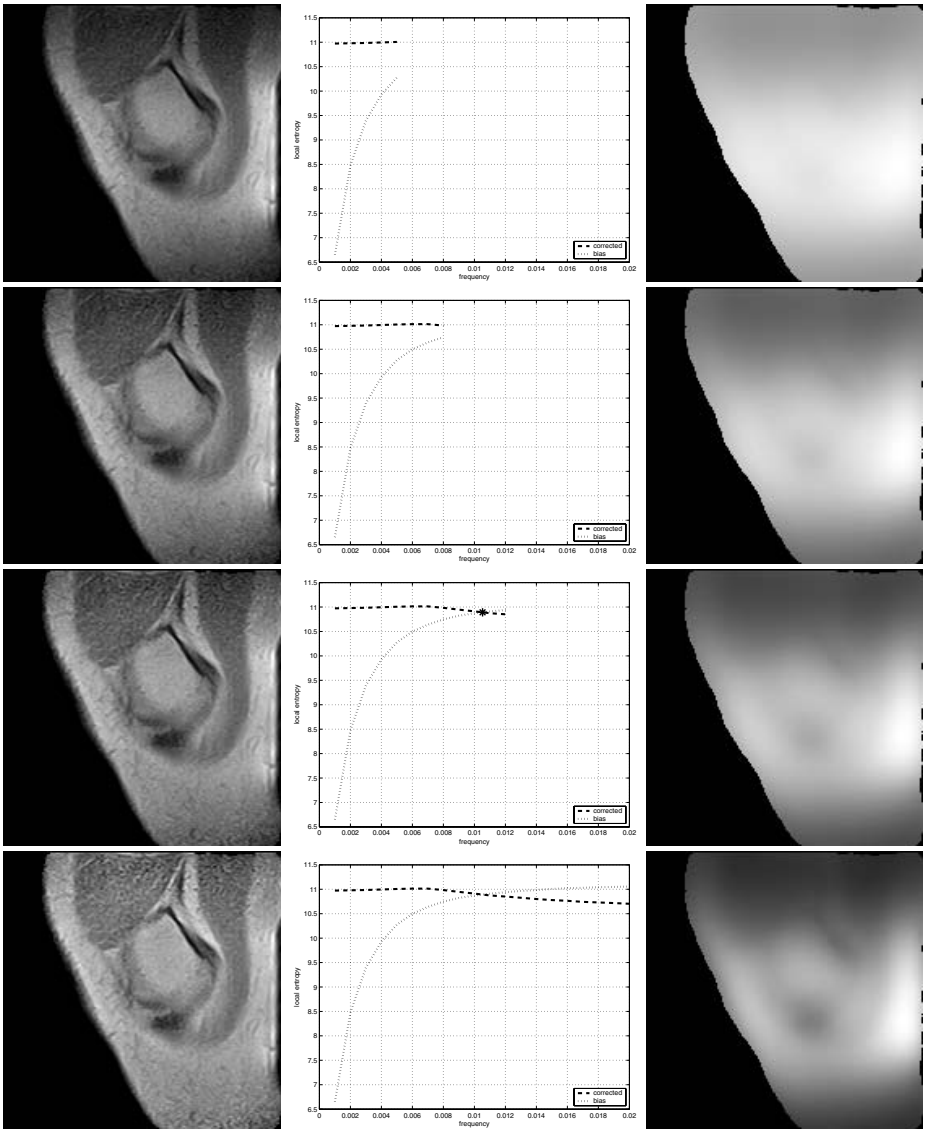


Fig. 2. Up-to-down: sequences at growing cut-off frequency. From Left-to-Right: corrected, diagram frequency-entropy,bias estimated.

3 Cut-Off Frequency Determination

From the information theory point of view, I has more information than necessary: it contains both not corrupted information and bias information, so the filtering can be considered like a sort of information transfer. It is necessary, then, introduce a function which measure the quantity of information which

is contained in an image: in correspondence of the ROI, it could be seen, according to a growing cut-off frequency,an increment of the information in I_{Bias}, while in $I_{corrected}$ should happen an information decreasing. The Shannon Entropy is a such information measure which has been already used in literature [6][13]:

$$H = -\sum p(x) \cdot \log_2 [p(x)] \tag{3}$$

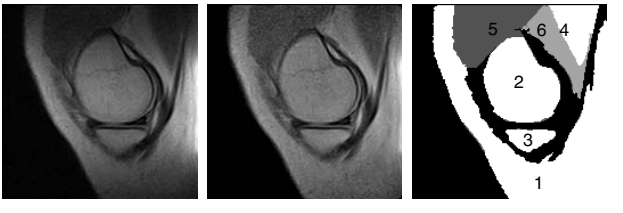
where x is the image histogram. In fig.2 this information transfer is shown. As it can be noticed, the corrected image has been chosen in correspondence of the intersection point of the two curves which are the local entropies both $I_{corrected}$ and I_{Bias}. The entropy is calculated only on ROI, because the background doesn't contain any information about the artifact.

4 Experimental Setup

The method has been performed on images decoded from DICOM file format, that is without any optical scanner acquisition which may introduce other artifacts. The device is an ESAOTE ARTOSCAN C with a magnetic field intensity of 0.18 Tesla .The dataset which we present in this paper is a complete study of the knee on sagittal plane which consists of 19 T1-weighted images acquired with the following parameters: Spin Echo sequence, Repetition time: 980 ms, Echo time: 26 ms, Slice thickness: 4mm, Flip Angle= 90. The useful resolution of FOV is 170x170 pixels with 12 bit of pixel depth. Both the original and restored images have been submitted to two physicians specialized in orthopaedy for a visual inspection. Only two restored images exhibit a gray level distortion, but they maintain the contrast among the tissues. All the restored images shows a panoramic illumination showing details on peripheral zone of the images, while in the original ones only the central zone of the anatomy is over illuminated. After the subjective evaluation performed above, the coefficient of variation has been performed on each zone showed in fig. 3

$$cv(zone) = \frac{\sigma(zone)}{\mu(zone)}$$

where σ and μ are, respectively, standard deviation and mean of the zone. As it can be seen in fig.4, the cv decrease in almost all the zones.



	coefficient of variation	
	before	after
zone1	0.492726	0.272752
zone2	0.165578	0.079895
zone3	0.107908	0.110375
zone4	0.319164	0.216040
zone5	0.272219	0.173506
zone6	0.332079	0.186072

Fig. 3. Form left to right:original image, restored image, manual segmentation, table of coefficient of variations

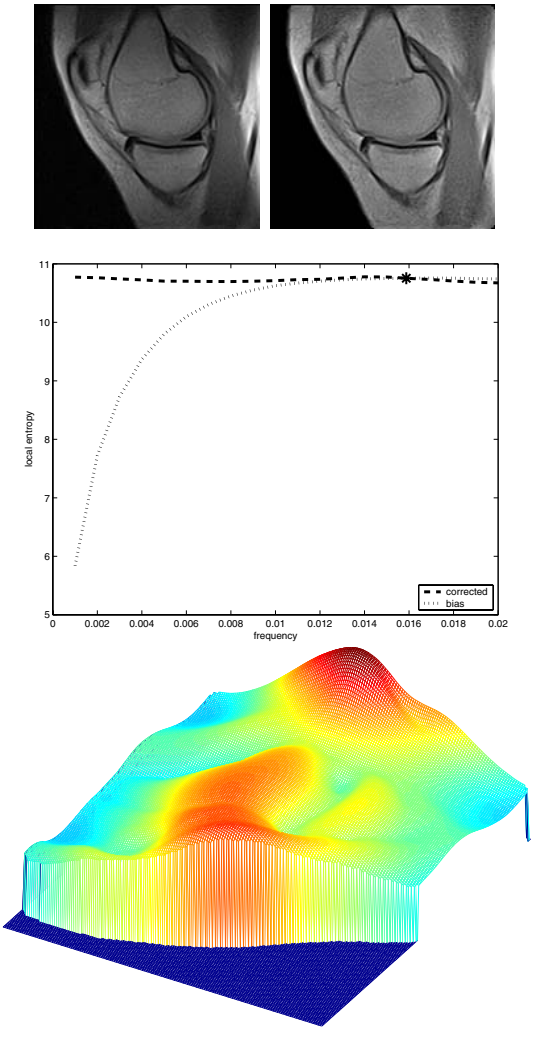


Fig. 4. Up to Down:original and restored images; local entropy diagram; Bias surface

5 Conclusions

An automatic technique has been presented to determine the cut-off frequency in homomorphic filtering for bias removal in MRI. The presented method exhibit several good properties:

- no hypothesis are made on tissue classes. It can be applied on wide kind of medical images; often the application presented in other papers is focused usually on encephalic MRI.
- Implementation is quite simple: it is an improvement of classical filtering approach in image analysis.

– There is no needs of preliminary experimental setup to "tune" the filter.
– The method performs on the fly and it is fully adaptive and automatic.

The method can be implemented in semi-automatic way in order to allow the doctor to set a preferred setting, so the cut-off frequency is only suggested by the algorithm. This is not a marginal fact: the doctor uses visual inspection of the image with the aim of to determine the disease.

Acknowledgement

This work has been partially supported by Istituto Radiologico PIETRO CIGNOLINI - Policlinico dell'Universitá di Palermo. Particulars thanks Dr. Daniele Peri for his technical support, Dr. Gian Piero De Luca and Dr. Claudio Cusumano for their medical support and prof. Giuseppe De Maria for his availability.

References

1. Styner, M.; Brechbuhler, C.; Szckely, G.; Gerig, G.: Parametric estimate of intensity inhomogeneities applied to MRI Medical Imaging. IEEE Transactions on Medical Imaging **22** (2000)153–165
2. Shang-Hong Laia; Ming Fangb: A dual image approach for bias field correction in magnetic resonance imaging. Magnetic Resonance Imaging **21** (2003)121–125
3. Mohamed N. Ahmed; Sameh M. Yamany; Nevin Mohamed: A Modified Fuzzy C-Means Algorithm for Bias Field Estimation and Segmentation of MRI Data. IEEE Transactions on Medical Imaging **21** (2002) 193–199
4. Van Leemput K.;Maes F.; Vandermeulen D. and Suetens P.: Automated Model-Based Bias Field Correction of MR Images of the Brain. IEEE Transactions on Medical Imaging **18** (1999) 885–896
5. Guillemaud, R.: Uniformity Correction with Homomorphic filtering on Region of Interest. IEEE International Conference on Image Processing **2** (1998) 872–875
6. Guillemaud, R.; Brady M. : Estimating the Bias Field of MR Image. IEEE International Conference on Image Processing **2** (1998) 872–875
7. Dawant B.M.; Zijdenbos A.P.; Margolin R.A.: Correction of Intensity Variations in MR Images for Computer-Aided Tissue Classification. IEEE Transactions on Medical Imaging **12** (1993) 770–781
8. Axel L.; Costantini J.; Listerud J.: Intensity Correction in Surface Coil MR Imaging. American Journal on Roentgenology **148** (1987) 418–420
9. Lei Jiang,Wenhui Yang:A Modified Fuzzy C-Means Algorithm for Segmentation of Magnetic Resonance Images. Proc. VIIth Digital Image Computing: Techniques and Applications.Sun C., Talbot H., Ourselin S. and Adriaansen T. Editions. (2003) 225–231
10. Tincher M.; Meyer C.R.; Gupta R.; Williams D.M.: Polynomial Modelling and Reduction of RF Body Coil Spatial Inhomogeneity in MRI. IEEE Transactions on Medical Imaging **12** (1993) 361–365
11. Brinkmann B. H. , Manduca A. and Robb R. A.: Optimized Homomorphic Unsharp Masking for MR Grayscale Inhomogeneity Correction. IEEE Transactions on Medical Imaging. **17** (1998) 161–171

12. Wells W.M.; Grimson W.E.L.; Kikins R.; Jolez F.A.: Adaptive Segmentation of MRI Data. IEEE Transactions on Medical Imaging. **15** 429–442 (1996)
13. Likar B.; Viergever M.A.; Pernus F.: Retrospective Correction of MR Intensity Inhomogeneity by Information Minimization. IEEE Transactions on Medical Imaging **20** (2001) 1398–1410
14. Pham D.L.; Prince J.L.: Adaptive Fuzzy Segmentation of Magnetic Resonance Images. IEEE Transactions on Medical Imaging. 18(9), (1999) 737-752
15. Pham D.L.; Prince J.L.: An Adaptive Fuzzy C-Means Algorithm for Image Segmentation in the Presence of Intensity Inhomogeneities. Pattern Recognition Letters. 20(1), (1999) 57-68
16. Gonzalez R.C.; Woods R.E.: Digital Image Processing. Prentice Hall Ed.
17. Arnold JB; Liow J-S; Schaper KS; Stern JJ; Sled JG; Shattuck DW; Worth AJ; Cohen MS; Leahy RM; Mazziotta JC; Rottenberg DA. Quantitative and Qualitive Evaluation of Six Algorithms for Correcting Intensity Non-Uniformity Effects. Neuroimage (2001) 13(5) 931–943.

Hybrid Surface- and Voxel-Based Registration for MR-PET Brain Fusion

Ho Lee[1] and Helen Hong[2,*]

[1] School of Electrical Engineering and Computer Science, Seoul National University
holee@cglab.snu.ac.kr
[2] School of Electrical Engineering and Computer Science BK21: Information Technology, Seoul National University, San 56-1 Shinlim 9-dong Kwanak-gu, Seoul 151-742, Korea
hlhong@cse.snu.ac.kr

Abstract. In this paper, we propose a novel technique of registration using hybrid approach for MR-PET brain image fusion. Hybrid approach uses merits of surface- and voxel-based registration. Thus, our method measures similarities using voxel intensities in MR images corresponding to the feature points of the brain in PET images. Proposed method selects the brain threshold using histogram accumulation ratio in PET images. And then, we automatically segment the brain using the inverse region growing with pre-calculated threshold and extract the feature points of the brain using sharpening filter in PET images. In order to find the optimal location for registration, we evaluate the Hybrid-based Cross-Correlation using the voxel intensities in MR images corresponding to the feature points in PET images. In our experiments, we evaluate our method using software phantom and clinical datasets in the aspect of visual inspection, accuracy, robustness, and computation time. Experimental results show that our method is dramatically faster than the voxel-based registration and more accurate than the surface-based registration. In particular, our method can robustly align two datasets with large geometrical displacement and noise at optimal location.

1 Introduction

In medical field, image fusion [1] is very useful for early diagnosis as well as for understanding the accurate location of disease by visualizing both anatomical and functional information simultaneously. For example, PET (positron emission tomography) image providing functional information detects changes in the metabolism caused by the growth of abnormal cells before anatomical abnormality, whereas MR (magnetic resonance) image providing anatomical information detects an accurate location of disease due to high resolution. Thus MR-PET image fusion can accurately interpret the location and range of disease with combined information. However, in order to fuse MR-PET brain images, image registration [1-7] determining the relationship of correspondence between two images with different resolution, position, and orientation is necessary.

* Corresponding author

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 930 – 937, 2005.

The registration methods used to fuse MR-PET brain images can be classified as moment-, surface-, or voxel-based registration. A.P.Dhawan et al. proposed moment-based registration [2]. This method respectively segments the brain from MR and PET images, and computes the center of gravity and its principal axes using zeroth and first order moments with the segmented brain. Registration is then performed by aligning the center of gravity and the principal axes. Since the computation of center of gravity is usually not accurate in blurry ones like PET images, this method is used primarily as a coarse pre-registration. Surface-based registration [3-4] requires delineation of corresponding surfaces in each of the images separately. Y.Hata et al. and L.Y.Hsu et al. proposed the methods to align of the brain surface extracted by automatic segmentation algorithms in MR and PET images. However, these approaches are not easy to accurately segment the only brain in MR images because its intensities are similar to the surrounding non-brain ones. Voxel-based registration [5-7] measures the similarity of all geometrically corresponding voxel pairs within overlapping parts. R.P.Woods et al. proposed a method based on the minimization of the sum of the standard deviation of PET voxel intensities corresponding to narrow ranges of MR voxel intensities [5]. F.Maes et al. proposed mutual information (MI) [7] measuring statistical dependence of intensities in corresponding regions of the two images. However, these methods require enormous processing time due to measuring similarity with all voxel pairs.

In this paper, we propose a novel technique of registration using hybrid approach for aligning MR-PET brain images. Hybrid approach that combines merits of surface- and voxel-based registration does not attempt to segment the brain from MR images, but only segments the brain in PET images, and then uses the Hybrid-based Cross-Correlation (HCC) to locate the feature points of PET images in the low intensities of voxel between the brain and the surrounding tissue of MR images. Thus, our method aligns MR and PET images by minimizing the similarity of the HCC that uses voxel intensities in MR images corresponding to the feature points of brain in PET images. In our experiments, we evaluate visual inspection, accuracy, robustness, and computation time with software phantom and clinical datasets.

The organization of the paper is as follows. In Section 2, we discuss our registration using hybrid approach in detail: the selection of brain threshold, the automatic segmentation of brain, the extraction of feature points within brain, and the similarity measure and optimization process to find exact geometrical relationship in MR and PET images. In Section 3, experimental results show how the method accurately, robustly, and rapidly aligns in software phantom and clinical datasets. This paper is concluded with a brief discussion of the results in Section 4.

2 Rigid Registration Using Hybrid Approach

Fig. 1 shows the pipeline of our method for the registration of MR and PET images. Since MR images have more anatomical information than PET images, MR images are fixed as reference volume and PET images are defined as floating volume. Our method selects the brain threshold using histogram accumulation ratio in PET images. And then, we automatically segment the brain using the 2D inverse region growing (IRG) with pre-calculated threshold value and extract the feature points of the brain

using sharpening filter in PET images. The feature points of PET images are transformed to MR images during iterative alignment procedure. At this time, in order to find optimized parameters we evaluate the HCC using the voxel intensities in MR images corresponding to the feature points in PET images. Interpolating PET images at grid positions of MR images is also required for the each iteration depending on the transformation. Since rigid transformation is enough to align the brain base, we use three translations and three rotations about the *x*-, *y*-, *z*- axis. Finally, aligned MR-PET images are displayed by a conventional volume rendering technique.

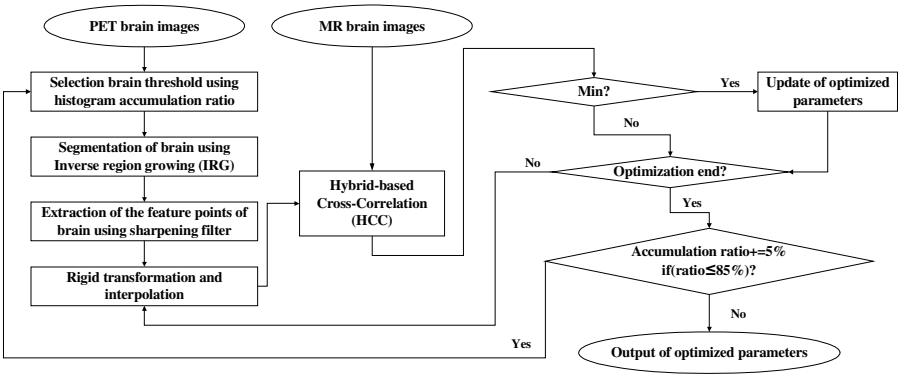


Fig. 1. The pipeline of our method using hybrid approach

2.1 Selection of the Brain Threshold Using Histogram Accumulation Ratio

Although the brain area of PET images has higher intensities than non-brain areas such as background and the surrounding tissue, it is difficult to accurately divide into the boundary between the brain and non-brain areas due to blurry properties of PET images. To allow an easy distinction between these areas, we use a histogram accumulation ratio that measures the ratio of non-brain volume of the whole volume.

The computation of ratio accumulates till the highest intensity beginning from the lowest intensity in histogram. If the ratio reaches a certain value, the background area having low intensities is included at first. If it is more accumulative, the surrounding tissue areas are included. At this time, the excluded areas are regarded as brain area, and we select the brain threshold by computing the average of intensity at these areas like following formula (1).

$$Threshold\ of\ brain = \sum_{i=k}^{n} \frac{H(i)\cdot i}{H(i)} \quad \left(if \quad Ratio(\%) = \frac{\sum_{i=0}^{k} H(i)}{\sum_{i=0}^{n} H(i)} \times 100(\%) \geq Init(\%) \right) \tag{1}$$

Here, H(i) is the measure of the number of certain intensity i in histogram, k is a intensity when the ratio, or Ratio(%), is more than or equal to a certain value as Init(%), and n is the maximal intensity of the whole intensities. The brain threshold is then selected by averaging from k to n.

Fig. 2 is the results applying threshold selected by histogram accumulation ratio when the ratio is increased every 5% from 50% to 85% in PET images. We can select the brain area and a little surrounding tissue when the ratio is 50%. When it is 85%, brain area is eroded.

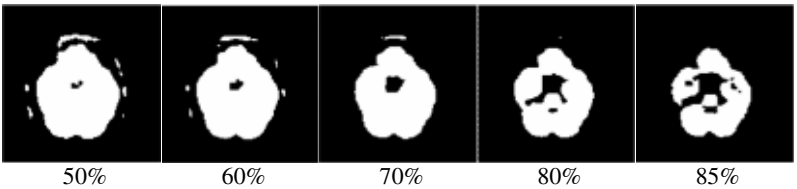


Fig. 2. The results of brain threshold selected by histogram accumulation ratio

2.2 Segmentation of the Brain Using 2D IRG and Extraction of Feature Points

Since the segmentation of brain using threshold-based method [8] can produce holes within the brain, these holes should be filled by morphological operations such as dilation and erosion [8]. However, morphological operations should determine the number of iteration in proportion to the size of holes. In addition computation time is increased by the number of iteration, and the numerous iterations can produce distortion of edge. We propose the 2D IRG for the automatic brain segmentation without these limitations in PET images.

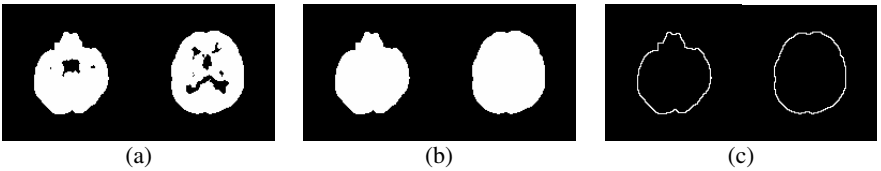


Fig. 3. Results of the segmentation of brain and the extraction of feature points in each PET slice (a) Threshold-based method (b) Proposed 2D IRG method (c) Feature points of the brain

The 2D IRG starts by choosing a seed pixel at (0, 0) on image and compares it with neighboring pixels. Region is grown from the seed pixel by adding neighboring pixels that are less than threshold selected by histogram accumulation ratio. When the growth of region stops, this region is non-brain areas. Then we simply segment the brain area by inverse operation. Thus the 2D IRG can automatically segment the brain area without inner holes and the distortion of edge by morphological operations in PET images. Fig. 3 shows the results of threshold-based, proposed 2D IRG method, and the feature points of the brain in the each PET slice. In Fig. 3(a), we can easily see inner holes within the brain, whereas proposed 2D IRG method can clearly segment the brain area without inner holes like Fig. 3(b). Fig. 3(c) shows the feature points of the brain extracted by applying a conventional sharpening filter [8].

2.3 Similarity Measure and Optimization

The similarity measure is used to determine the degree of resemblance of windows in successive frames. Most of voxel-based similarity measures take very long processing time because of considering the voxel intensities of whole volume. We propose the Hybrid-based Cross-Correlation (HCC) that uses the only voxel intensities in MR images corresponding to the feature points in PET images. Our approach reduces computation time because of using the only voxel intensities corresponding to the feature points instead of the voxel intensities of whole volume. The HCC can be defined as formula (2).

$$HCC = \frac{1}{N_{PET}} \sum_{i=0}^{N_{PET}} V_{MR}(Tr(P_{PET}(i))) \tag{2}$$

Here, N_{PET} is total number of feature points in PET images. $P_{PET}(i)$ is the position of *i*-th feature point in PET images. Tr is rigid transformation matrix transforming feature points of PET images into the coordinate system of MR images. V_{MR} is voxel intensities in MR images corresponding feature points in PET images. In order to search for the optimal location, we find optimized six parameters such as T_x', T_y', T_z', R_x', R_y', R_z' when the HCC reaches minimum as following formula (3).

$$Tr(T_x', T_y', T_z', R_x', R_y', R_z') \leftarrow min(HCC) \tag{3}$$

The both images are initially positioned such that their center coincides and that the corresponding scan axes of both images are aligned and have the same orientation. Powell's multidimensional direction method [7] is then used to minimize the HCC. This method searches for optimal location in the order following T_x, T_y, R_z, R_x, R_y, T_z until the HCC doesn't change any more and iterates over constant number. In this paper, the number of iteration is fixed to five because of converging without update of optimized parameters.

3 Experimental Results

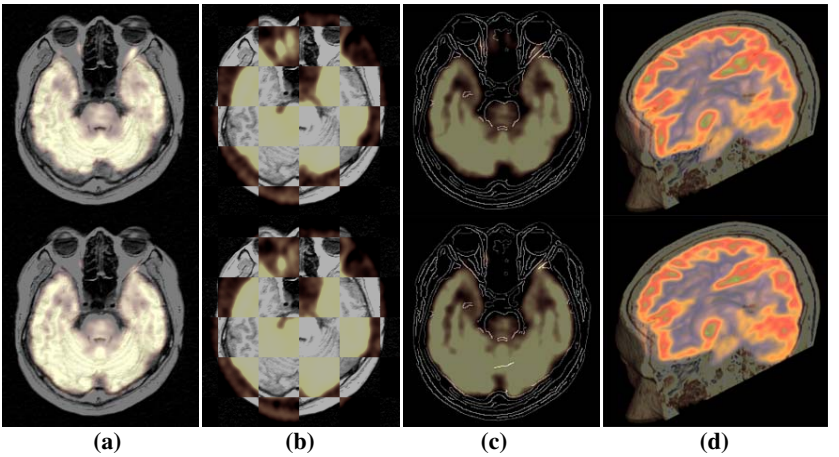
All our implementation and test were performed on an Intel Pentium IV PC containing 3.0 GHz CPU and 1.0 GBytes of main memory. Our method has been successfully applied to two clinical datasets and two software phantom datasets as described in Table 1. PET software phantom is generated by dividing into three area such as background, tissue, and brain. The Gaussian smoothing is also added for blurry properties. CT software phantom is generated by dividing into four areas such as background, tissue, brain, and skull. The Gaussian noise is also added to CT of software phantom2 for robustness test. The performance of our method was evaluated with the aspects of visual inspection, accuracy, robustness, and computation time.

For visual inspection in clinical datasets, we provided different display methods such as frame-based, checkerboard-based, edge-based, and three-dimensional data mixing and rendering. As the top row of Fig. 4 is to apply scale parameters only before registration in arbitrary slice, we can see that the brain area of PET images

misaligns on that of MR images. Whereas we can confirm that the brain area is correctly aligned in the bottom row of Fig. 4 applying optimal parameters. In Fig. 4(d), we can see the inner brain area by cutting on arbitrary 3D.

Table 1. Experimental datasets

Dataset	MR/PET	Volume size	Voxel size	Intensity range
Patient1	MR-T1	256×256×94	0.90×0.90×1.50	-1024 ~ 3091
	FDG-PET	128×128×40	1.95×1.95×3.33	0 ~ 255
Patient2	MR-T2	256×256×76	0.78×0.78×1.62	-1024 ~ 3091
	FDG-PET	128×128×90	2.00×2.00×2.00	0 ~ 4095
SW-Phantom1	MR	256×256×94	0.30×0.30×0.3	-1024 ~ 3091
	PET	128×128×40	0.32×0.32×0.3	0 ~ 4095
SW-Phantom2	MR	256×256×96	1.00×1.00×1.00	-1024 ~ 3091
	PET	128×128×48	2.00×2.00×2.00	0 ~ 4095



(a) **(b)** **(c)** **(d)**

Fig. 4. The results of visual inspection before and after registration (a) frame-based display (b) checkerboard-based display (c) edge-based display (d) 3D fusion

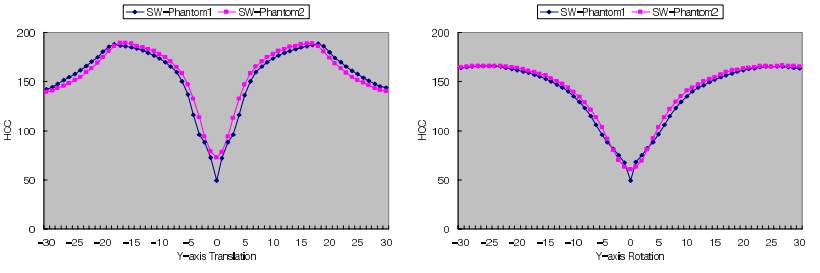
For accuracy test, we evaluated the root-mean-square-error (RMSE) [8] as formula (4) using the arbitrary transformed software phantom datasets. Then, we compared the RMSE of our method with the one of surface- and voxel-based registration as Table 2. Surface-based registration achieved alignment by measuring correlation between the feature points of the brain extracted by semi-automatic segmentation tools. Voxel-based registration used mutual information method. Our method could keep the quality of the result of voxel-based registration and give more accuracy than the surface-based registration. In Table 2, translation unit is voxel, and rotation one is degree.

$$RMSE = \sqrt{\frac{1}{6}\left\{(T_x - T_x')^2 + (T_y - T_y')^2 + (T_z - T_z')^2 + (R_x - R_x')^2 + (R_y - R_y')^2 + (R_z - R_z')^2\right\}} \quad (4)$$

Table 2. Comparison of RMSE between proposed method and conventional methods

Arbitrary transformed parameters		Optimized parameters			
			Proposed	Surface-based	Voxel-based
T_x	-2.4	T_x'	-2.375	-4.125	-2.438
T_y	-3.6	T_y'	-3.563	-1.813	-3.500
T_z	-0.5	T_z'	-0.563	0.000	-0.438
R_x	-2.1	R_x'	-2.188	0.000	-2.100
R_y	1.0	R_y'	1.000	0.000	1.000
R_z	-4.0	R_z'	-4.000	-1.688	-4.000
RMSE			0.047796	1.69186	0.050478

For robustness test, we evaluated whether the HCC searches for optimal location in software phantom datasets with a large geometrical displacement and noise. As shown in Fig. 5, we can confirm that our method searches for optimal location where the HCC is a minimum at 0 voxel or degree. In particular, our result robustly converged on optimal location even in software phantom dataset2 added Gaussian noise.



(a) (b)

Fig. 5. Robustness test (a) Translation in the y-direction (b) Rotation in the y-direction

The total computation time is summarized by comparing our method with the conventional surface- and voxel-based registration in Table 3. Our method was dramatically faster than the voxel-based registration and was suitable for clinical use.

Table 3. Comparison of computation time between our method and conventional methods

(sec)

	Proposed method	Surface-based	Voxel-based
Patient1	23.8	3.3	648.1
Patient2	43.1	7.2	1755.7
SW-phantom1	24.2	3.4	639.8
SW-phantom2	25.4	4.1	721.5

4 Conclusion

We have developed a novel technique of registration using hybrid approach that combines merits of surface- and voxel-based registration for MR-PET brain fusion. Our histogram accumulation ratio was suitable for the selection of threshold in blurry ones such as PET images. The 2D IRG automatically segmented the brain without inner holes in PET images. In particular, the HCC could reduce computation time because of considering the only voxel intensities of MR images corresponding to the feature points extracted from PET images. Experimental results showed that our method was dramatically faster than the voxel-based registration and more accurate than the surface-based registration. Moreover, our method could robustly align two datasets with large geometrical displacement and noise at optimal location.

Acknowledgements

This work was supported in part by a grant B020211 from Strategic National R&D Program of Ministry of Science and Technology and a grant 10014138 from the Advanced Technology Center Program. The ICT at Seoul National University provides research facilities for this study.

References

1. J.B.A.Maintz, M.A.Viergever, A survey of medical image registration, Medical Image Analysis, Vol.2, Iss.1 (1998) 1-36.
2. A.P.Dhawan, L.K.Arata, Alejandro V.Levy, and Joseph Mantil, Iterative Principal Axes Registration Method for Analysis of MR-PET Brain Images, IEEE Transactions on Biomedical Engineering, Vol.42, No.11 (1995) 1079-1087.
3. Y.Hata, S.Kobashi, S.Hirano, and M.Ishikawa, Registration of Multi-Modality Medical Images by Soft Computing Approach, ICONIP 6th International Conference on Neural Information Processing, Vol.3 (1999) 878-883.
4. L.Y.Hsu, M.H.Loew, Fully automatic 3D feature-based registration of multi-modality medical images, Image and Vision Computing Vol.19 (2001) 75-85.
5. R,P,Woods, J.C.Mazziotta, S.R.Cherry, MRI-PET registration with automated algorithm, Journal of Computer Assisted Tomography, Vol.17 (1993) 536-546.
6. J.Cizek, K.Herholz, S.Vollmar, R.Schrader, J.Klein, W-D, Heiss, "Fast and robust registration of PET and MR images of human brain," Neuroimage, Vol.22, Iss.1 (2004) 434-442.
7. F.Maes, A.Collignon, G.Marchal, P.Suetens, Multimodality Image Registration by maximization of Mutual Information, IEEE Transaction on Medical Imaging, Vol.16, No.2 (1997) 187-198.
8. R.G.Gonzalez, R.E.Woods, Digital Image Processing, 1st Ed. (1993).

A System to Support the Integrated Management of Diagnostic Medical Images

Andrea F. Abate, Rosanna Cassino, Gabriele Sabatino, and Maurizio Tucci

Dipartimento di Matematica e Informatica,
Università di Salerno,Via Ponte don Melillo,
84084 Fisciano, Salerno – Italy
{abate, rcassino, gsabatino, mtucci}@unisa.it

Abstract. Information systems are essential tools supporting the management of hospital organizations. The demand for availability and integration of data in this field is more and more increasing, basically for absolving two key issues: collecting and merging all the elementary data available for a patient during the hospitalization process, so that physicians and other operators get all the necessary information they need during the process; planning the development of the diagnostic activities/therapeutics to optimize the process. In this perspective, we present a system that integrates a booking subsystem for hospital specialized treatments booking (CUP), a subsystem for the management of the hospitalizations (including First Aid Departments), a subsystem for filing and reporting clinical images, a subsystem for the analysis of radiological images in a unique management environment. Therefore we describe a complete system for the management of an archive of digital dossiers for the patients of a hospital, where diagnostic imaging is a central issue.

1 Introduction

The diffusion of informative systems in the field of sanitary firms is very heterogeneous, due to the complex organization of these structures, often articulated in a number of operational units, departments and services, each of them having ample decisional autonomy. Another reason for heterogeneity often lies in the limited budget for the ICT investments that has hindered a univocal approach to the demands of computerizing.

This fragmentation has accordingly determined a proliferation of informative systems, heterogeneous both in architectural terms (hardware platforms, software technologies and nets), and in applicable and functional terms (interfaces, instructions and processes). So many separate "islands" have been created that they do not even communicate among them. The need of integration of systems and information is now becoming more apparent to the medical and nursery staffs themselves, as they are experimenting increasing difficulties in matching today's management objectives using old technologies and traditional tools [10].

The role of the Departments of Diagnostic Radiology is particularly relevant in the development of management systems supporting the diagnostics processes, as the Ra-

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 938 – 944, 2005.

diological Information Systems (RIS) are assuming a leading identity within the Hospital Information Systems in the definition of requirements and objectives and in the evaluation of innovative technical solutions.

Typical database solutions to manage patient related data (acceptance, scheduling, filing, statistics) are usually implemented in the existing RIS systems, but tools for the integrated management of diagnostic imaging in PACS systems are often still lacking, so that many Radiology Departments are still storing physical diagnostic images although more efficient and less expensive digital storing solutions are now available and in many cases allowed by specific laws and regulations [12].

In this perspective, in this paper we present an integrated system to manage both the administrative information of the patients and the achievement and analysis of the related medical images. This work was developed within a project founded by the Italian Minister of Health Care and developed by the "Girolamo Orlando" Foundation of Pescopagano in collaboration with the Department of Mathematics and Computer Science at Salerno University. The goal of this project was the functional integration of a traditional management system for hospital treatment booking (CUP), hospital admission management and medical reporting with a diagnostic imaging system [1, 2].

The paper is organized as follow. Section 2 describes the proposed system architecture underlying the characteristics of the integration between the several involved systems. In section 3 we present the implementation characteristics of the integrated environment. Finally, in section 4 we show some conclusion and any further work.

2 System Architecture

In this paper we present a case study related to the development of an integrated system to manage the administrative information of the patients and the achievement and analysis of the related medical images. The developed prototype intends a series of objective: produce new tools to aid the physician for the analysis of the images and for the formulation of the diagnosis through the native comparison of images to known diagnosis; create a given basic representative of the expertise, that is a Knowledge Basic Management System of the diagnosis, of the treatment protocols, etc.; provide a support to the didactic; make available the expertise in the net, create a database for possible recourses legal physician. In the integration process between the several applications – all object oriented - the processes of interaction and exchange of the data has stayed standardized and conformed to the more recent specifics in terms of safety and of efficiency. Therefore, all the exchanges occur by web services, e the process to receive or send the data by XML/HL7 (v.2.3.1) [9], in domain ports logic, respectively (applicative, delegate). Such specific allows using only the web type net protocol (http, https) and then an impact reduced on the configuration of possible net devices. Particularly, the communication foresees that the receiving entity exposes a web services, called from the transmitting entity for the dispatch of the data.

Figure 1 presents a general description of the flows that interest all the evolution phases in which a contact could be generally dismountable.

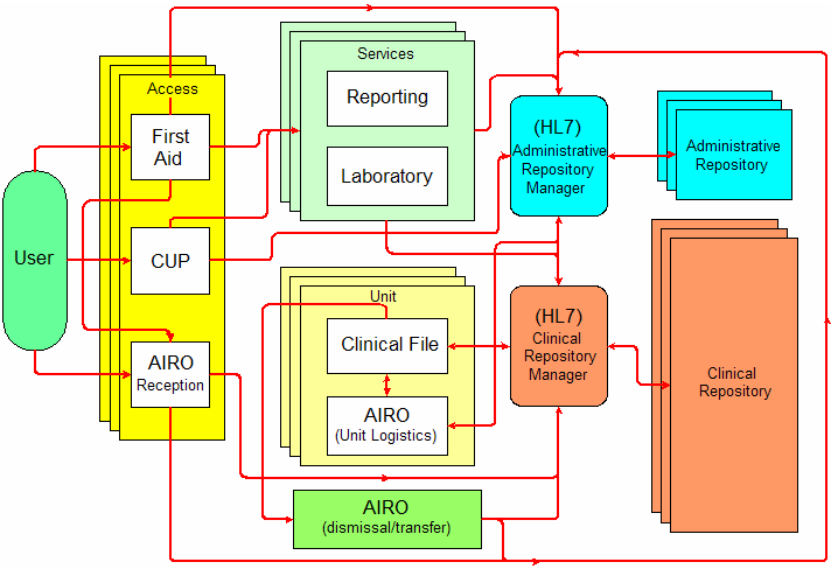


Fig. 1. Flows of the information of a patient to contact with a sanitary structure

The informative sanitary system taken to reference is based essentially on the integration of the functionalities which a whole of procedures each of which appointed to the management of specific appearances of the system offers. All the procedures refer to the "Administrative Repository" and to the "Clinical Repository" database. The "Administrative Repository" contains the registry information of the patients that have had any contact with the sanitary structures. The "Clinical Repository" contains the digital clinical files, the reports and the links to the medical images that are in the database of the servers where such pictures are achieved and related to the other information of the patients. The new reporting environment allows the inside interaction with the CUP processes (related to the bookings of specialized performances) with the systems for the management of the activities of First Aid and of the Departments (AIRO - hospitalization informative area) and finally with all the tools to analyze and process the clinical images.

The problems related to the management of the clinical images could be outlined through a process of analysis and diagnosis composed of four separate phases: acquisition and elaboration of the images; management of the data and of the images; diagnosis formulation; classification and filing of the information. Each phase is not isolated but is support and verify for the others. Figure 2 shows the applicative process model that describes the potentialities of the proposed prototype in the complex. The extreme versatility of the prototype allows wide integration with all the other present systems in the hospital structures beyond that the possibility from part of all the authenticated user from the system to access to a large field information, by means of a simple internet connection, service staying the presence of the tricks to safeguard the sensitive data and to respect the privacy.

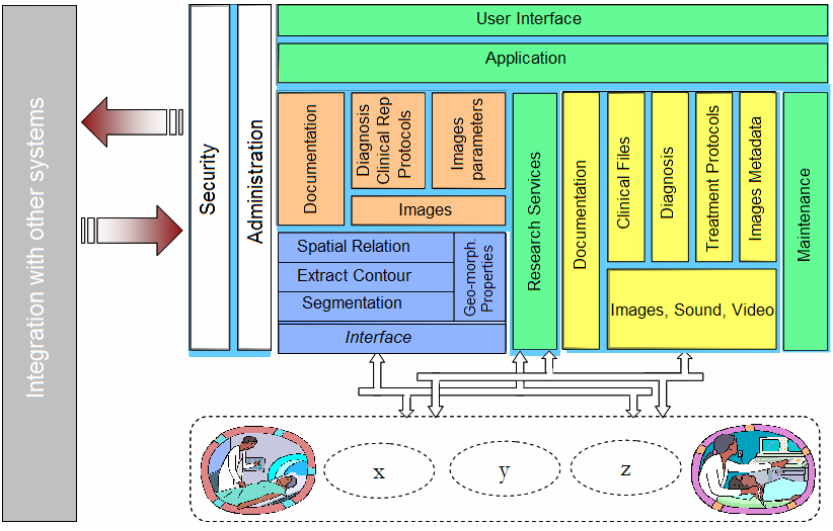


Fig. 2. The applicative process model

The clinical images elaboration phases, after they have stayed achieved according to DICOM 3 protocol [11] are:

- *segmentation*,
- *extract contour*,
- evaluation of the *geo-morphological properties,*
- *spatial relations* computing.

If the image contains some abnormality whose geo-morphological characteristics are automatically extracted within the corresponding spatial relations. In this matter, the analysis of a pathology proceeds with the calculus of its geo-morphological characteristics and on the basis of the spatial arrangement of these last ones. All such tasks are performed using different algorithms already available in the literature. In detail, the analysis of a clinical image segments the images, underlines the contours of the individualized abnormality (segmentation [8] and edge detection phases [6]), computes the geo-morphological parameters (information related to the density of the analyzed object, asymmetry, orientation, spreadness, uniformity, convexity [3, 4, 5]) and, by means of the spatial relationship between the objects (particularly the representative objects of the abnormality and the surrounding anatomical components) describes the *virtual image* [7] related to the examined image. Then, after the clinical images of a patient are achieved, these will be analyzed inside the environment of the described system. The physician will mail the criterions of search and visually formulate queries that extract from the database all the images similar to the analyzed images on the base of the matching of the geo-morphological parameters understood in a determined range of values of reference. The physician verifies the drawn out contents and analyzes the agreement with regards to his interest. In the case the system has extrapolated objects coherent with the abnormality to diagnose, he could recall all the documentation correlated to the image (administrative and clinical data and above all the reports already issued), so to be supported in the new diagnosis phase.

All useful information to the updating of the clinical file of the patient are organized on the base to their content for then have filed correctly and represented graphically. The classified data are: the administration documentation, the patient personal data, the diagnosis, the treatment protocols, the clinical files, the images parameters, (see figure 2). The categorized operations allow to feed the database and, on the base of the instruction of the user about the informative grade of the sample document to which he has had access, it is possible optimize the documents presentation. Thanks to the presence of a thematic engine, through the web, it is possible to agree to all the information of the database. It represents a real "Knowledge Base Management System", useful for the physician, furnishing aid to the analysis of the images and to the diagnosis formulation, for the Hospital Institutes present on the territory, allowing the expertise in the net, and for the Universities Institute, constituting a formidable information field supporting didactic.

3 System Implementation

As stated before, the aim of this work was integration between a management system for patient administrative information and a system achieving, analyzing and processing related medical images.

The first system is implemented in .NET environment using Microsoft SQL Server whereas the second uses Java and an Oracle database.

As a first step, we planned to use web services technology as a standard for interoperability between different environments. After a services oriented analysis, we pointed out that physicians are less comfortable with different windows on desktop especially during diagnosis report so we decided porting all systems in .NET environment using SQL Server as DBMS. We used J# to migrate Java code in .NET environment.

Inside the subsystem devoted to diagnostic imaging are implemented the processes of segmentation and edge detection of the individuated pathology described in previous section (Fig. 3).

Once the contour of the abnormality located in the examined radiological image is extracted, the analysis of pathology proceeds with the calculus of its geomorphological characteristics. This information will be used for querying the database relatively to the similarity retrieval for geo-morphological characteristics of other images previously examined.

The proposed Visual Data Base Management System is based on integration of icon based methodology used in GIS systems with similarity retrieval techniques in the field of diagnostic imaging systems. This helps the diagnosis process of physicians that could use a retrieval technique in order to access other documents into the Database. There is also the possibility to express a diagnosis using a template including all information related to medical examination and annotations if necessary.

Such a process could be used even if physician is interested in analyzing an examination or comparing with similar ones.

From such point of view, the whole process is a completion of application-functional aspect that allows the systems to manage the digital dossier of a patient and his clinical history.

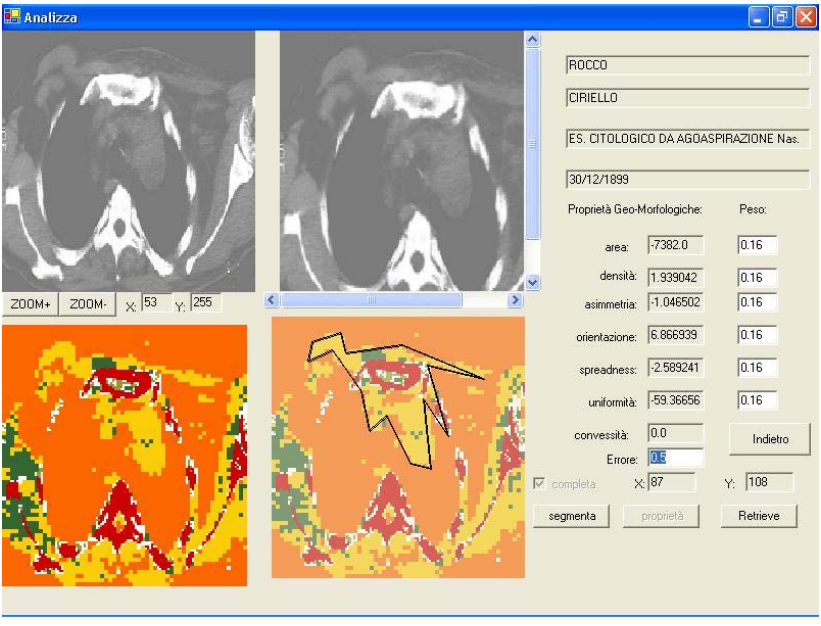


Fig. 3. Segmentation and geo-morphological properties of an image

4 Conclusion and Further Work

The system illustrated in the previous sections supports hospital centers in the integrated management of every patient-related information available during a treatment plan, including diagnostic imaging. The main feature of this application is flexibility, which lies in its ability in customizing the user interface for any particular department's needs.

Currently, the system has been proposed to different sanitary firms like the hospital “S. Carlo” in Potenza (Italy) and the Azienda Sanitaria Locale (ASL) n. 2 in Venosa, Potenza (Italy). Future work will focus on using the results of the present case study evaluation to predict the innovation impact of introducing an integrated RIS/PACS system in these real hospital contexts. Preliminary tests on the presented application have indeed shown that the availability of integrated information effectively enhances a RIS/PACS system's support of diagnostic activities.

Further evolution of this work will consider new image processing functions to manipulate and manage diagnostic images, especially to provide physicians with 3D-based visualization for image analysis and retrieval.

References

[1] Abate A. F., Cassino R. and Tucci M.: An Integrated Environment for Control and Management of Pictorial Information Systems. Proceeding of First International Workshop "Image 2003: Learning, Understanding, Information Retrieval, Medical", World Scientific Press, Vol. 15, (2003) 123-131
[2] Abate A. F., Nappi M., Tortora G., Tucci M.: IME: An Image Management Environment with Content-based Access, Image and Vision Computing, Elsevier Science, Vol. 17, (1999) 967-980
[3] Bryant A.: Recognizing Shapes in Planar Binary Images. Pattern Recognition, Vol. 22, (1989) 155-164
[4] Dunham J. G.: Optimum Uniform Piece Linear Approximation of Planar Curves. IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI, Vol. 8, n. 1, (1986)
[5] Gritzali F., Papakonstantinou G.: A Fast Piece Linear Approximation Algorithm, Signal Processing, Vol. 5, (1983) 221-227
[6] Papakonstantinou G..: Optimal Polygonal Approximation of Digital Curves. Signal Processing, Vol. 8, (1985) 131-135
[7] Petraglia G., Sebillo M., Tucci M., Tortora G.: Virtual Images for Similarity Retrieval in Image Databases, IEEE Transactions on Knowledge and Data Engineering, Vol. 13, n. 6, (2001), 951-967
[8] Vitulano S., Di Ruberto C., Nappi M.: Different methods to segment biomedical images, Pattern Recognition Letters, Vol. 18, (1997)
[9] http://www.hl7.org
[10] http://www.microsoft.com/italy/biztalk/
[11] http://medical.nema.org/
[12] http://www.teinos.com/ita/ris_pacs_it.html

Volume Estimation from Uncalibrated Views Applied to Wound Measurement

B. Albouy[1], S.Treuillet[2], Y. Lucas[1], and J.C. Pichaud[3]

[1] Laboratoire Vision & Robotique, Université d'Orléans
ENSI, 10 Bd Lahitolle 18000 Bourges, France
Benjamin.Albouy@ensi-bourges.fr,
Yves.Lucas@bourges.univ-orleans.fr
[2] Polytech'Orléans, Site Galilée, 12 rue de Blois, BP 6744, 45067 Orléans, France
Sylvie.Treuillet@univ-orleans.fr
[3] Hôpital La Tour Blanche, Issoudun, France
jc.pichaud@wanadoo.fr

Abstract. The aim of the ESCALE project is to supply the medical staff with objective and accurate 2D and 3D measurements for wound healing assessment from color images acquired in a free manner with a low cost digital camera. The problem addressed in this paper is the volume estimation from uncalibrated views. We present two experimentations. A Monte Carlo simulation on synthetic perturbated data leads to an average error of 3% on reconstructed points. Then, triangulation based volume estimation obtained from two uncalibrated real images gives us hope that an accuracy less than 5% is achievable. So this technique is suited to accurate wound 3D measurements. Combined with true color image processing for colorimetric tissue assessment, a such low cost system will be appropriate tool for diagnosis assistance and therapy monitoring in clinical environment.

1 Introduction

Pressure ulcers or bed sores, are damages of the skin affecting people confined to bed or wheelchair, exposed to prolonged pressure on bony prominence, such as the buttocks and heels. These wounds may appear in a few hours but the healing process can last from two weeks (30% of cases) to several years. Whether pressure sores occur or not depends on patient related factors. The most important of these factors are age, neurological condition, dietary status, blood circulation and skin moisture degree. Several surveys reveal an average of prevalence between 15% and 20% among hospitalized patients. Within the E.U., at least 2 million persons suffer from chronic skin wounds, such as pressure ulcers or diabetic lesions. Health care costs attributed to chronic skin wounds are estimated at 8 billions euros per year and the number of ulcers is expected to increase by over 25% by the year 2010, as a consequence of the growing part of aged population.

The effective management of non healing wounds is based on a complete patient history, a detailed initial assessment of the wound, and an analysis of probable causative factors. This information is used to personalize a management strategy to the underlying pathophysiology preventing healing and to implement appropriate

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 945 – 952, 2005.

wound interventions. Regular reassessment of healing progress and medical care changes are also necessary. In [1], the authors review clinically useful wound measurement approaches, and encapsulate key wound parameters in the simple mnemonic, which begins by *MEASURE*: length, width, depth, and area. But quantitative and accurate wound measurements remain a substantial challenge, because clinical practices are currently limited to qualitative visual evaluation [2]. Sometimes wound area measurement is carried out from acetate manual tracing, more rarely alginate moulds are used for measuring volume. All these methods are painful for the patient and subject to contamination, irritation or allergic reaction by a contact with the wound site. Following upon this lack of effective metric to evaluate the wound healing, the investigation and progress in wound treatment are limited. Hence, there is a demand from the medical staff for a convenient, accurate, reproducible and non-contact method to achieve wound periodic assessment and therapeutic monitoring.

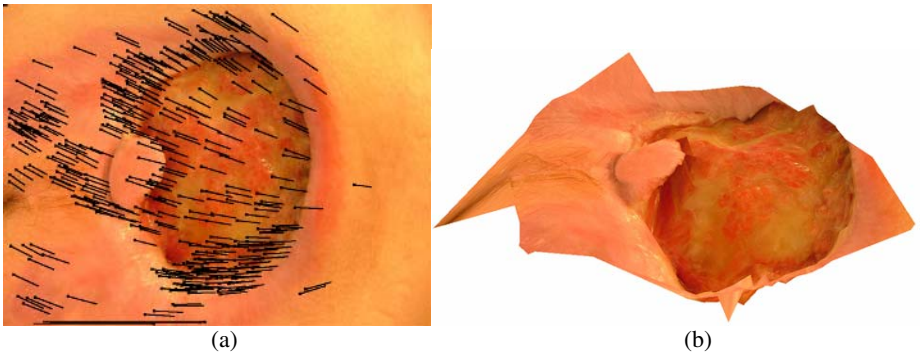
1.1 Relative Works

To address this problem, a variety of techniques based on image processing have been explored since ten years [3-10]. The first works started from single pictures and carried out multi-center clinical studies to compare wound area measurements obtained by computerized imaging techniques (digitalized pictures) and acetate tracings. Another way investigates color analysis for skin healing assessment and kinetics [3,4]. But 2D measurements are not sufficient for a complete wound assessment [1]. Because wound healing frequently occurs through changes in depth rather than surface area, 3D measurements are necessary.

Plassmann and Jones [5, 6] pioneered a structured light technique with a system called *MAVIS* (Measurement of Area and Volume Instrument System): a set of color-coded stripes is projected onto the wound area and recorded by a CCD camera at an angle about 45 degrees. They used a plaster model to achieve measurements with a mean error of less than 5% (if the wound area is greater than 9 cm^2 and less deep than 3 cm). Nevertheless, the precision of results from wound models, undertaken as part of a validation trial of the tool, could not achieved during clinical practice. However, *MAVIS* was consistently more precise than other methods used, it was noted to have high precision in wounds with larger dimensions, but it was of no value in large, circumferential, deep or under-mining wounds. Krouskop [9] used also structured light grid pattern of 16x16 dots captured by a digital camera to calculate the area and volume of wound. Image processing takes less than 5 minutes for a trained computer operator. Volume precision has been estimated on plaster molds with spherical indentation. Results indicate a precision within ±3%, when at least 144 of pattern dots lie within the wound.

A second approach was developed with *MEDPHOS* (Medical Digital PHOtogrammetric System) in [7, 8]: three cameras are mounted on a triangular frame in a slightly convergent way. The set-up includes also additional light source and texture projection in the center. The distance between cameras is fixed to 15 cm and implies a distance to the wounds of about 30 cm. The calibration step is critical for the accuracy of the digital model and involves a trained operator. Experiments with patients were carried out on different types of wounds. This method sometimes failed due to many specular reflections caused by moisture on the wound surface, and the

main goal of next works has been to improve the robustness. Finally, *DERMA* system [10] uses a Minolta VI910 scanner to acquire the 3D geometry of the wound and to capture a RGB image aligned to the geometry with 640x480 pixels resolution. In conclusion, we can observe that all previous systems are expensive and cumbersome equipment, not really suitable in clinical environment. Furthermore these dedicated vision systems fail in measuring very large wounds and their lack of portability is a severe drawback for imaging not easily accessible wounds without uncomfortable position for the patients.



(a) (b)

Fig. 1. The 3D color model of a real wound may be obtained from two uncalibrated views. (a) Left view with about four hundred inliers selected by MAPSAC method, (b) Color texture mapped on 3D model obtained by Delaunay triangulation.

1.2 Escale Project

The aim of ESCALE project is to develop image processing integrated tools to provide accurate 2D and 3D measurements of wounds from a series of color pictures (Fig.1). Unlike previous works, we have favored a portable, fast and safe technique for the patient. So, we prefer to use a standard digital camera to provide the medical staffs with a handy low cost image acquisition system. Digital camera is now a widespread device which delivers very high resolution color images (3 Mpixels or more), with zoom facility to adapt the field of view on a large variety of wound dimensions (from 4 to 400 cm^2). The portability is also useful for tele-care purpose with home nursed patients by internet image transfer.

Behind this easy and free image capture, real technical difficulties are dissimulated, such as unknown geometry views and varying lighting conditions. So, we develop advanced techniques concerning 3D uncalibrated reconstruction and color segmentation. The problem addressed in this paper is the volume estimation from uncalibrated views. The subject of 3D reconstruction from a series of images is a widely studied topic in computer vision community. But what about the final triangulation accuracy that could be obtained on 3D points from two uncalibrated views? And what volume precision could we hope for? This subject is not really documented because uncalibrated reconstruction is generally devoted to 3D visualization or augmented reality from video sequences [11]. The reconstruction chain implemented is described in the next section. A Monte Carlo simulation provides error estimation to compare the influence of the fundamental matrix

estimation algorithms in 3D point reconstruction. Next, the volume estimation obtained from triangulation on real images is presented with experimental results in section 3. Finally, we conclude and propose further improvements.

2 Reconstruction Error from Uncalibrated Views

Two images taken from different points of view allow the reconstruction of 3D positions thanks to a triangulation algorithm. But without information on geometry between the two views, a self-calibration step is needed before to run a triangulation algorithm. The 3D reconstruction chain starts with point correspondences on the two images. Several thousands of interest points (corners) are detected in each color image by the Harris & Stephen algorithm adapted by Gouet [12]. As usual, this detection is corrupted by noise. Next, point matching is realized by cross correlation using a local similarity function based on textured neighbourhood of each corner. Unfortunatly, this process provides a lot of correspondence mismatches. These outliers may be extracted by applying MAPSAC method [13]. This method computes the fundamental matrix from a random set of seven corner correspondences and tests all other matches against the epipolar constraint. A number of consistent correspondences is obtained by applying a threshold distance to epipolar lines. This step is repeated for many random sets and the matrix obtaining the largest number is accepted. The outcome is a set of inliers (Fig.1a). The fundamental matrix is then replaced by its closest rank-2 matrix by using singular value decomposition before to apply self-calibration based on Kruppa equations [14].

To simplify the computation of the intrinsic parameters, we apply some common assumptions (squared pixels and same focal length for all the views). Then, extrinsic parameters are calculated and it is possible to reconstruct the 3D scene using a classical triangulation method. But, as the geometry is only defined up to a scale factor, one distance must be known on the real world to achieve Euclidian measurements on the reconstructed data. Performance characterization of this reconstruction chain can be accessed in [15]. To estimate the error on 3D points reconstruction, a Monte Carlo simulation involves an iterative process using randomly noised data inputs. Several hundred of 3D points are randomly generated and projected in the two image plans under a known geometry. To model the effect of noisy image detection, Gaussian noise has been added on the image projections. Mismatches between images have also been simulated.

Five fundamental matrix estimation algorithms have been compared in the global reconstruction chain presented above (Fig.2). It appears that 8-points method is the most robust for noisy image inputs, and that MAPSAC is the most robust for mismatches. So, the two methods have been mixed: MAPSAC is used to classify good and bad correspondences, whereas 8-points algorithm is used to compute the fundamental matrix from resulting inliers. On these synthetic data, 3D reconstruction mean error is less than 3% with regards to a good triangulation configuration (60 degree angle between the two views). Applying this reconstruction chain on real images allow to obtain triangulation and color texture mapped 3D model (Fig.1b). Pictures are compressed and the performance of this algorithm under JPEG image degradation has already been tested by Torr [16]. It is suggested that $Q = 70$ is the most an image should be compressed without risk of substantial degradation. So using

compressed images is not a problem. Volume estimation obtained from triangulation on real images is presented in next section.

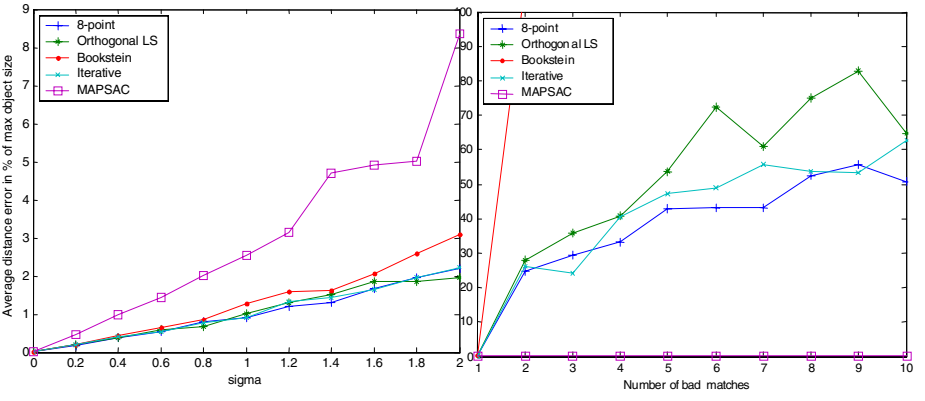


Fig. 2. Performance evaluation of fundamental matrix estimation algorithms by a Monte Carlo simulation. The y-axis gives the average distance error between reconstructed points and reference data, relative to (a) standard deviation of the Gaussian noise added on positions of projected points (b) number of mismatches with a standard deviation of Gaussian noise fixed to unity ($\sigma = 1$).

3 Triangulation Based Volume Estimation

Delaunay triangulation is the leading method for generating surface from 3D points. This triangulated surface may also be used for volume estimation [17]. The volume calculation consists in summing the elementary volumes under each prism formed by facets and their orthogonal projections on a reference plane. This plane, P, corresponding to the complete healing of healthy skin, is the mean plane in least square sense computed from points on the wound bounds. The equation for P is calculated in the camera reference frame. Once this plane is known, a rigid transformation H is applied to reconstructed data in order to fit the reference plane with the canonic plane P_{ref}: $z = 0$. So, the definition of H is $P_{ref} = k H \cdot P$, where k is a scale factor due to homogeneous coordinates. The rigid transformation H includes two rotations around x-axis and y-axis, and a translation t_z along z-axis. These four parameters (θ_x θ_y t_z k) are determined by analytical equations.

In order to achieve measurements with a mean error, repetitive tests have been done on a cork wound model over several pairs of images from different points of view (Fig.3). A pair of colored balls with known distance is placed in the field of view to provide a metric reference. These balls are automatically detected in the images. The contour of the wound is roughly drawn by manual selection. The ground truth is given by the mean of twenty successive weighting the water inner filling on a balance with a centigram precision. Volume estimations have been realized on sixteen pairs of images. Results are reported in Table 1. Comparison of theses estimations against the ground truth is given in Table 2. The triangulation based volume is slightly overestimated with a percentage error around 3%.

Even if this reconstruction accuracy is sufficient for wound volumetric estimation, these preliminary results raise two difficulties. The first one is the uncertainty of reconstruction. It highly depends on the angle between the triangulation rays that is the movement between the images. Points are less precisely localized along the ray as the rays become more parallel (forward motion in particular). The reconstruction error is large for high noise level, since there is a little movement between images. Uncertainty of measurements may be noticed in Table 1: the standard deviation to mean ratio rises up to 13%. The best configuration is obtained for a 90 degrees angle between views. But in this case, it will be very difficult to achieve reliable matching. This explains why published research [11] is based mostly on video sequences to gradually go over distant points of views through successive close images. A second difficulty derives from singularities that could lead to an insolvable self calibration [18]. This is the case for orbital movement (pure rotation by going round the object), for example. Further attention needs to be given to the image capture protocol, in particular by taking more than two views.

Table 1. Volume estimation on sixteen pairs of images

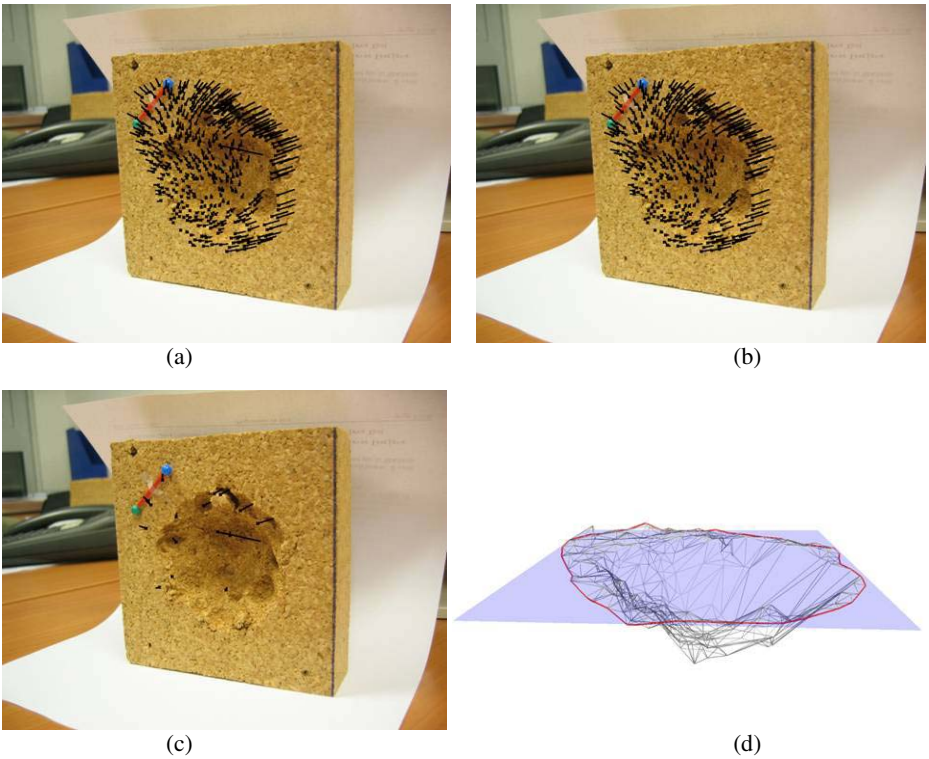
Measurements	44.40	41.70	34.60	44.27	40.71	40.58	43.90	33.80
	41.60	40.61	30.86	37.43	54.15	40.63	39.68	40.37

Table 2. Comparison of the triangulation based volume estimation against ground truth

	Mean	Standard deviation	Min	Max	St. dev./Mean
Estimated volume	40.58	5.25	30.85	54.15	12.93%
Ground truth	39.20	1.03	37.38	41.10	2.63%

4 Conclusion and Future Works

For a vision module to be of practical use for volume measurements, its performance must be evaluated. This paper present two experimentations. First a 3% error rate is observed on reconstructed points by a Monte Carlo simulation on synthetic perturbated data. Next, a volume estimation based on uncalibrated triangulation on real images suggests that an accuracy less than 5% may be achievable. So this technique is suited to clinical wound 3D measurements. Taking into account the natural curvature of the body in the 3D surface of a healing wound will increase the quality of volume estimation. Several improvements are currently implemented for the matching step, a major difficulties in image processing, like using geometry invariants and relaxation. In order to be of general practical use over a wide variety of conditions and to avoid self calibration singularities, more views will be used to refine triangulation. Combined with true color image processing for quantitative colorimetric tissue assessment, such a low cost system will be a convenient tool for diagnosis assistance and therapy monitoring in clinical environment.



(a) (b)

(c) (d)

Fig. 3. A cork model of wound is used to estimate the accuracy of triangulation based volume calculation**.** (a) All matches obtained on image, (b) inliers, (c) outliers, (d) triangulated surface with wound bound and reference plan.

Acknowledgement

This research was supported by the french Ministry of Research and the European Social Found through their delegations in Centre area.

References

1. Keast D.H., & al.: MEASURE: A proposed assessment framework for developing best practice recommendations for wound assessment. Wound Repair and Regeneration, June Vol.12 (2004) s1-s17
2. Pudner Rosie: Measuring wounds. (2002) http://www.jcn.co.uk/
3. Herbin M., Bon F.X., Venot A., Jenlouis F., Dubertret M.L.: Assessment of healing kinetics through true color image processing. IEEE Transactions on Medical Imaging 12-1 (1993) 39-43
4. Bon F.X., Briand E., Guichard S., Couturaud B., Reval M., Servant J.M. and Dubertret L.: Quantitative and kinetic evolution of wound healing through image analysis. IEEE Transactions on Medical Imaging 19-7 (2000) 767-772

5. Plassmann P., Jones B.F., Ring E.F.J.: A structured light system for measuring wounds. Photogrammetric Record Vol.15 N°86 (1995) 197-203
6. Plassman P., Jones T.D.: MAVIS: a non-invasive instrument to measure area and volume of wounds. Med. Eng. Phys 20-5 (1998) 332-338
7. Boersma S.M., Van den Heuvel F.A., Cohen A.F., Scholtens R.E.M.: Photogrammetric wound measurement with a three-camera vision system. IAPRS Vol. XXXIII Amsterdam (2000)
8. Malian A., Heuvel Van Den F.A., Azizi A.: A Robust Photogrammetric System for Wound Measurement. ISPRS Symposium commission V Corfu Greece International Archives of Photogrammetry and Remote Sensing Vol. 34 Part 5 (2002) 264–269
9. Krouskop T.A, & al.: A non contact wound measurement system. Journal of Rehabilitattion R&D Vol. 39 N°3 May/June (2002) 337-346
10. Romanelli M., & al.: Technical advances in wound bed measurement. WOUNDS (2002) 14 –58
11. Pollefeys M. : Self-Calibration and Metric 3D Reconstruction From Uncalibrated Image Sequences. PhD Thesis of the Catholic University of Leuven (1999)
12. Gouet V.: Mise en correspondance d'images en couleurs, application à la synthèse de vues intermediaires, Thèse de Doctorat de l'Université de Montpellier II Sciences et Techniques du Languedoc (2000)
13. Torr P.H.: Bayesian Model Estimation and Selection for Epipolar Geometry and Generic Manifold Fitting. International Journal of Computer Vision 50-1 (2002) 35-61
14. Sturm P.: On Focal Length Calibration from Two Views. Conference on Computer Vision and Pattern Recognition, vol. II (2001) 145-150
15. Albouy B., Treuillet S., Lucas Y. Birov, D.: Fundamental matrix estimation revisited through a global 3D reconstruction framework. IEEE ACIVS Bruxelles (2004) 185-192
16. Torr P.H, Zisserman A.: Performance characterization of fundamental matrix estimation under image degradation. Machine Vision & Application Vol.9 (1997) 321-333
17. De Floriani K., Puppo E.: An on-line algorithm for Delaunay triangulation. Graphic Models and Image processing Vol.54 N°3 (1992) 290-300
18. Sturm P. : Vision 3D non calibrée : Contributions à la reconstruction projective et études des mouvements critiques pour l'auto-calibrage. Thèse de Doctorat au Laboratoire GRAVIR – IMAG de l'INRIA Rhône-Alpes (1997)

Scatter Search Particle Filter for 2D Real-Time Hands and Face Tracking

Juan José Pantrigo, Antonio S. Montemayor, and Raúl Cabido

Universidad Rey Juan Carlos, c/ Tulipán s/n
28933 Móstoles, Spain
{juanjose.pantrigo, antonio.sanz}@urjc.es, rcabido@gmail.com

Abstract. This paper presents the scatter search particle filter (SSPF) algorithm and its application to real-time hands and face tracking. SSPF combines sequential Monte Carlo (particle filter) and combinatorial optimization (scatter search) methods. Hands and face are characterized using a skin-color model based on explicit RGB region definition. The hybrid SSPF approach enhances the performance of classical particle filter, reducing the required evaluations of the weighting function and increasing the quality of the estimated solution. The system operates on 320x240 live video in real-time.

1 Introduction

Automatic visual analysis of human motion is an active research topic in Computer Vision and its interest has been growing during the last decade [1]. Human-computer interaction is going to non-contact devices, using perceptual and multimodal user interfaces. That means the system allows the user to interact without physical contact, using voice or gesticulation capture [2]. The gesture tracking by monocular vision is an important task for the development of many systems. Recently, the field of computer vision has devoted considerable research effort to the detection and recognition of faces and hand gestures [3]. To locate the regions of interest, this kind of systems needs for previous multiple objects tracking procedure. Tracking human movement is a challenging task which strongly depends on the application [4].

Recent research in human motion analysis makes use of the particle filter (PF) framework. The particle filter algorithm enables the modelling of a stochastic process with an arbitrary probability density function (pdf), by approximating it numerically with a set of points called particles in a state-space process [5]. Unfortunately, particle filter fails in high dimensional estimation problems such as articulated objects [6] or multiple object tracking [7]. Hands and head tracking is an instance of multiple object tracking, involving a six-dimensional state space. In these kind of problems, particle filters may not be enough.

The main contribution of this paper is the development of a multiple object visual tracker based on the scatter search particle filter (SSPF) algorithm [8]. The scatter search (SS) metaheuristic proposed by Glover [9][10] is a population based metaheuristic applied in the context of combinatorial optimization problems. SSPF hybridizes both PF and SS frameworks in two different stages. In the PF stage, a

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 953 – 960, 2005.

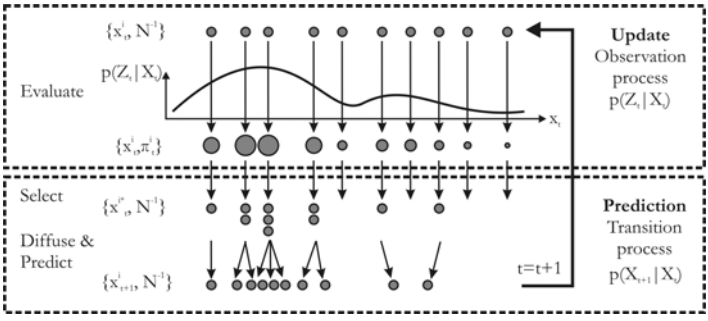


Fig. 1. Particle Filter scheme

particle set is propagated and updated to obtain a new particle set. In the SS stage, an elite set from the particle set is selected, and new solutions are obtained by combining them. SSPF significantly improves the performance of particle filters. This algorithm has been effectively applied to real-time hands and face tracking. Results show the system operates in real-time on a standard PC computer.

The rest of the paper is organized as follows. Next section presents the basic particle filter framework. Section 3 shows the scatter search procedure itself while section 4 describes the hybridization of scatter search and particle filter (called SSPF). Section 5 and 6 demonstrates its application to a 2D face and hands tracking problem. Finally, we present our conclusions and future works in section 7.

2 Particle Filter Framework

Sequential Monte Carlo algorithms (also called particle filters) are filters in which theoretical distributions on the state space are approximated by simulated random measures (called particles) [5]. The state-space model consists of two processes: (i) an observation process $p(\boldsymbol{Z}_t|\boldsymbol{X}_t)$, where $\boldsymbol{X}$ denotes the system state vector and $\boldsymbol{Z}$ is the observation vector, and (ii) a transition process $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})$. Assuming that observations $\{\boldsymbol{Z}_0, \boldsymbol{Z}_1,\ldots, \boldsymbol{Z}_t\}$ are known, the goal is to recursively estimate the posterior pdf $p(\boldsymbol{X}_t|\boldsymbol{Z}_t)$ and the new system state $\{\chi_0, \chi_1, \ldots, \chi_t\}$ at each time step. In Sequential Bayesian Modelling framework, posterior pdf is estimated in two stages:

(i) Evaluation: posterior pdf $p(\boldsymbol{X}_t|\boldsymbol{Z}_t)$ is computed at each time step by applying Bayes theorem, using the observation vector $\boldsymbol{Z}_t$:

$$p(\mathbf{X}_t \mid \mathbf{Z}_t) = \frac{p(\mathbf{Z}_t \mid \mathbf{X}_t)p(\mathbf{X}_t \mid \mathbf{Z}_{t-1})}{p(\mathbf{Z}_t)} \tag{1}$$

(ii) Prediction: the posterior pdf $p(\boldsymbol{X}_t|\boldsymbol{Z}_{t-1})$ is propagated at time step t using the Chapman-Kolmogorov equation:

$$p(\boldsymbol{X}_t \mid \boldsymbol{Z}_{t-1}) = \int p(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1})p(\boldsymbol{X}_{t-1} \mid \boldsymbol{Z}_{t-1})d\boldsymbol{X}_{t-1} \tag{2}$$

A predefined system model is used to obtain an updated particle set.

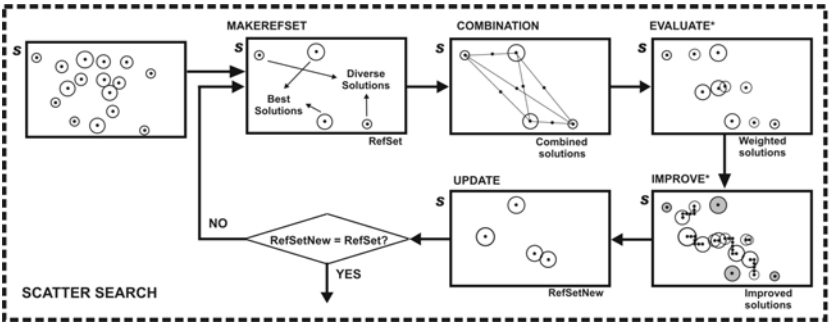


Fig. 2. Scatter Search scheme

In Fig. 1 an outline of the particle filter scheme is shown. The aim of the PF algorithm is the recursive estimation of the posterior pdf $p(\boldsymbol{X}_t|\boldsymbol{Z}_t)$, that constitutes the complete solution to the sequential estimation problem. This pdf is represented by a set of weighted particles $\{(\boldsymbol{x}_t^0, \pi_t^0)\ldots (\boldsymbol{x}_t^N, \pi_t^N)\}$.

PF starts by setting up an initial population $\boldsymbol{X}_0$ of N particles using a known pdf. The measurement vector $\boldsymbol{Z}_t$ at time step t, is obtained from the system. Particle weights $\boldsymbol{\Pi}_t$ are computed using a weighting function. Weights are normalized and a new particle set $\boldsymbol{X}^*_t$ is selected. As particles with larger weight values can be chosen several times, a diffusion stage is applied to avoid the loss of diversity in $\boldsymbol{X}^*_t$. Finally, particle set at time step t+1, $\boldsymbol{X}_{t+1}$, is predicted using the motion model. Therefore, particle filters can be seen as algorithms handling the particles time evolution. Particles move according to the state model and are multiplied or died according to their weights or fitness values as determined by the likelihood function [5].

3 Scatter Search

Scatter search (SS) [9][10] is a population-based metaheuristic that provides unifying principles for recombining solutions based on generalized path construction in Euclidean spaces. In other words, SS systematically generates disperse set of points (solutions) from a chosen set of reference points throughout weighted combinations. This concept is introduced as the main mechanism to generate new trial points on lines jointing reference points. SS metaheuristic has been successfully applied to several hard combinatorial problems.

In Fig. 2 an outline of the SS is shown. SS procedure starts by choosing a subset of solutions (called *RefSet*) from a set S of initial feasible ones. The solutions in *RefSet* are the h best solutions and the r most diverse ones of S. Then, new solutions are generated by making combinations of subsets (pairs typically) from *RefSet*. The resulting solutions, called trial solutions, can be infeasible. In that case, repairing methods are used to transform these solutions into feasible ones. In order to improve the solution fitness, a local search from trial solutions is performed. SS ends when the new generated solutions do not improve the quality of the *RefSet*.

4 Scatter Search Particle Filter

Visual tracking of articulated motion is a complex task with high computational costs. Due to the dynamic nature of the problem, sequential estimation algorithms are usually applied to visual tracking. Unfortunately, particle filter fails in high dimensional estimation problems such as articulated objects or multiple object tracking. These problems can be seen as *dynamic optimization problems*. In our opinion, dynamic optimization problems deal with optimization and prediction tasks. This assumption is supported by the fact that the optimization method for changing conditions needs from adaptive strategies. On the other hand, in dynamic optimization problems it is not good enough to predict, and high quality solutions must be found.

Scatter search particle filter (SSPF) integrates both SS and PF frameworks in two different stages:

- In the PF stage, a particle set is propagated over the time and updated to obtain a new one. This stage is focused on the evolution in time of the best solutions

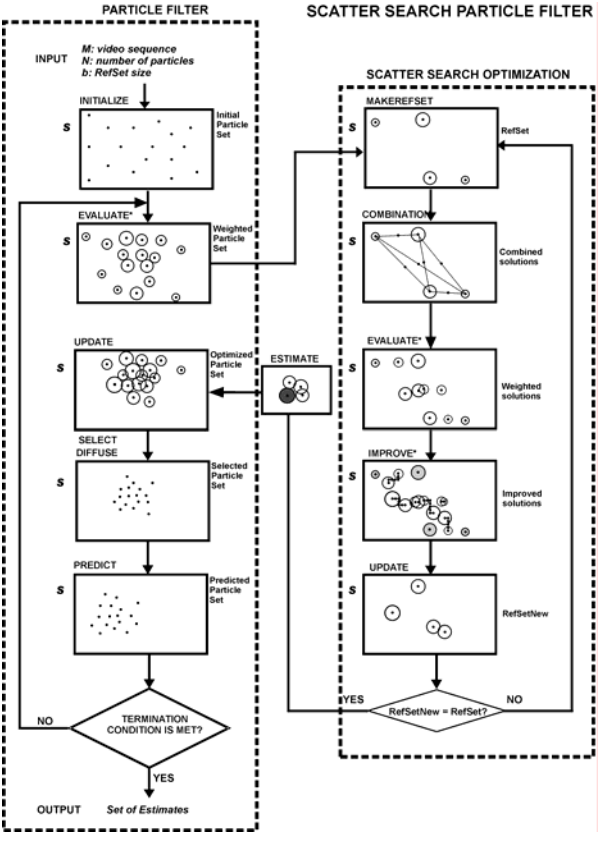


Fig. 3. Scatter Search Particle Filter scheme. Weight computation is required during EVALUATE and IMPROVE stages (*)

found in previous time steps. The aim for using PF is to avoid the loss of diversity in the solution set and to adapt the system to changing conditions.

- In the SS stage, a fixed number of solutions from the particle set are selected and combined to obtain better ones. This stage is devoted to improve the quality of a reference subset of good solutions in such a way that the final solution is also improved.

Fig. 3 shows a graphical template of the SSPF algorithm. Dashed lines separate the two main components in the SSPF scheme: PF and SS optimization, respectively. SPF starts with an initial population of N particles drawn from a known pdf (Fig. 3: INITIALIZE). Each particle represents a possible solution of the problem. Particle weights are computed using a weighting function (Fig. 3: EVALUATE). SS stage is later applied to improve the best obtained solutions of the particle filter stage. A Reference Set (*RefSet*) is created selecting a subset of b ($b<<N$) particles from the particle set (Fig. 3: MAKEREFSET). This subset is composed by the $b/2$ best solutions and the $b/2$ most diverse ones of the particle set. New solutions are generated and evaluated, by combining all possible pairs of particles in the *RefSet* (Fig. 3: COMBINE and EVALUATE). In order to improve the solution fitness, a local search from each new solution is performed (Fig. 3: IMPROVE). Worst solutions in the *RefSet* are replaced when there are better ones (Fig. 3: UPDATEREFSET). SS stage ends when new generated solutions *RefSetNew* do not improve the quality of the *RefSet*. Once the SS stage is finished, the "worst" particles in the particle set are replaced with the *RefSetNew* solutions (Fig. 3: INCLUDE). Then, a new population of particles is created by selecting the individuals from particle set with probabilities according to their weights (Fig. 3: SELECT and DIFFUSE). Finally, particles are projected into the next time step by following the update rule (Fig. 3: PREDICT).

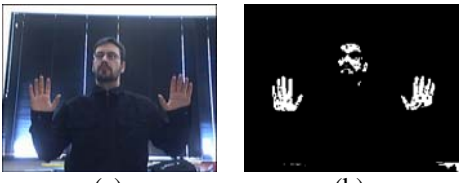
The SSPF leads the search process to a region of the search space in which it is highly probable to find new better solutions than the initial computed ones. SSPF search in state-space is not performed randomly like in a general particle filter. Unlike the PF algorithm, SSPF is time-adaptive since the number of evaluations of the weighting function changes in each time step.

5 SSPF Implementation to Hands and Face Tracking

Our proposed SSPF system is applied to determine the position of hands and face in 2D image sequences. Each particle in the particle set describes a possible solution for the tracking problem. The *particle structure* used in this work is:

$$\left[x_F, y_F, x_{RH}, y_{RH}, x_{LH}, y_{LH}, \dot{x}_F, \dot{y}_F, \dot{x}_{RH}, \dot{y}_{RH}, \dot{x}_{LH}, \dot{y}_{LH}\right] \tag{3}$$

where x and y are the spatial positions, and $\dot{x}$ represents the first derivative of magnitude (velocity). Subscripts F, RH and LH are referred to face, right and left hands respectively. The number of particles N in the particle set S is chosen to be 100. The *RefSet* is created by selecting the 6 best solutions in S.



(a) (b)

Fig. 4. Observation model: (a) input image, (b) skin color detection

The *observation model* specifies the image features to be extracted. To construct the weighting function it is necessary to use adequate image features. In this work we have used a pixel based skin color detection method (see Fig. 4b). In this method an explicit region of RGB color space is defined as skin. A pixel (R,G,B) is classified as skin if [11]:

$$\begin{aligned}(R>45)\,\&\,(G>40)\,\&\,(B>20)\,\&\,(\max(R,G,B)-\min(R,G,B)>15)\\ \&\,(|R-G|>15)\,\&\,(R>G)\,\&\,(R>B)\end{aligned} \tag{4}$$

The obvious advantage of this method is the simplicity of skin detection rules that leads to construction of a very rapid classifier. Particle weights are computed as the number of skin pixels belonging to three rectangular windows located in $[x_F, y_F, x_{RH}, y_{RH}, x_{LH}, y_{LH}]$.

System model describes how particles evolve over the time. The update rule used in this work is:

$$\begin{cases} x_{t+\Delta t} = x_t + \dot{x}_t \Delta t \\ \dot{x}_{t+\Delta t} = \dot{x}_t + G_x \Delta t \end{cases} \tag{5}$$

where x represents some spatial variable, Δt is the temporal step and G_X is a random Gaussian variable with zero mean and normal deviation σ_X.

The *combination procedure* consists of exploring all possible combinations using partitions of solutions. Lets consider two solutions *(F1, R1, L1)* and *(F2, R2, L2)*, where *F*, *R* and *L* represent the kinematic state of face and left and right hands respectively. Six new solutions *(F1, R2, L2)*, *(F2, R1, L1)*, *(F2, R1, L2)*, *(F1, R2, L1)*, *(F2, R2, L1)* and *(F1, R1, L2)*, are obtained by combining parts of solutions. Finally, the best solution is chosen as result of the combination procedure.

A standard *local search* was employed as an *improvement stage* in the SS scheme. Given a solution, a neighborhood is explored until a new high quality solution is found. Then, this new solution replaces the old one and the procedure is repeated until no improvement is produced.

6 Experimental Results

To analyze the performance of the proposed algorithm, implementations of PF and SSPF have been developed. The experiments were evaluated in a 1.7 GHz Pentium 4, 256 MB RAM under Microsoft Windows XP Home SP2. Real-time video capture and

Fig. 5. Hands and face tracking using (a) PF and (b) SSPF

processing have been done using Microsoft DirectShow API. In particular, the algorithms have been implemented as a DirectShow transform filter. The SSPF based system successfully operates on 320x240 live video in real time (28Hz). Fig. 5 shows the performance of the tracker using PF and SSPF. Fig. 6 shows the performance of SSPF vs. PF in the estimation of hands and head tracking during a video sequence. In this experiment the standard deviation using SSPF was dramatically lower than using PF as it can be seen in Table 1. These results demonstrate that the SSPF based approach increases the performance of general PF, improving the quality of the estimated solution and reducing the required evaluations of the weighting function. As results, the frame rate obtained using SSPF is higher than using PF, as it can be seen in Fig. 7.

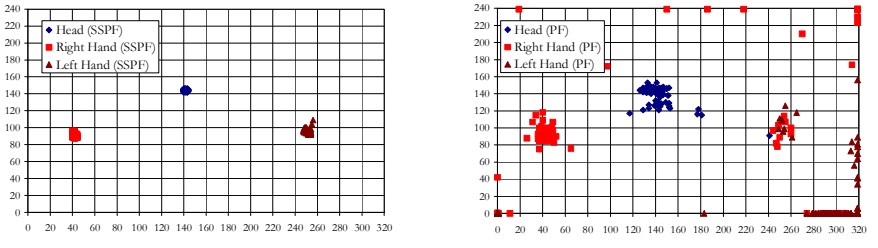


Fig. 6. Estimation of hands and head position using (a) SSPF and 282 particles and (b) PF and 400 particles, during 133 video frames for the static pose shown in fig 4.

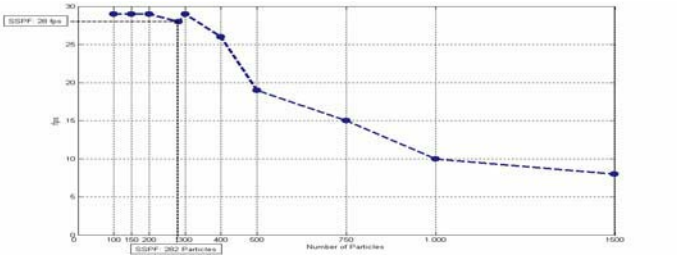


Fig. 7. Frame rate using different number of particles in PF and using SSPF

Table 1. Standard deviation (σ) for SSPF and PF for x and y coordinates during 133 video frames for the static pose shown in fig 4

Algorithm	N_{part}	*Head*		*Right Hand*		*Left Hand*	
		σ_x	σ_y	σ_x	σ_y	σ_x	σ_y
SSPF	282	1.1	1.1	1.2	1.7	1.8	2.1
PF	400	11.9	9.2	125.2	72.3	119.6	33.3
	1000	7.2	5.8	17.9	57.5	91.7	35.0

7 Conclusion

The main contribution of this work is the development of a 2D face and hands tracker based on the scatter search particle filter (SSPF) algorithm. SSPF hybridizes the scatter search metaheuristic and the particle filter framework to solve dynamic problems. Experimental results show how SSPF appreciably increases the performance of PF without losing quality in the estimation procedure. As a result, the tracker works in real time on a standard PC. Future works will deal with the management of occlusions and the correspondence problem to develop a vision based human-computer interface.

References

1. Wang, L., Weiming, H., Tieniu, T.: Recent developments in human motion analysis. *Pattern Recognition* 36, 585–601 (2003)
2. Buades, J.M., Perales, F.J., Varona, J. Real time segmentation and tracking of face and hands in VR Applications. *AMDO LNCS 3179*, 259-268 (2004)
3. MacLean, J., et al. Fast Hand Gesture Recognition for Real-Time Teleconferencing Applications. *In proc. of the 2nd Int Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems* (2001)
4. Argyros, A.A., Lourakis, M.I.A. Real time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera. *In proc. of the European Conference on Computer Vision (ECCV'04)*, Springer-Verlag, vol. 3, 368-379 (2004)
5. Arulampalam, M., et al.: A Tutorial on Particle Filter for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Trans. On Signal Processing, V 50 (2): 174–188 (2002)
6. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. IEEE Conf. Computer Vision and Pattern Recognition, Vol. 2 (2000) 126–133
7. C. Hue, J.-P. Le Cadre, P. Pérez. A particle filter to track multiple objects. In *IEEE Workshop on Multi-Object Tracking*, 61-68 (2001)
8. Pantrigo, J.J., Duarte, A., Sánchez, A.: Scatter Search Particle Filter to Solve the Dynamic Travelling Salesman Problem. *EVOCOP 2005 LNCS* 3448, 177-189 (2005)
9. Glover, F.: A Template for Scatter Search and Path Relinking. *LNCS*, 1363, 1-53 (1997)
10. Laguna, M., Marti, R.: Scatter Search methodology and implementations in C. Kluwer Academic Publisher (2003)
11. Peer, P., Kovac, J., Solina, F. Human skin colour clustering for face detection. *In proc. of the InternationalConference on Computer as a Tool EUROCON* (2003)

Skin Detection in Videos in the Spatial-Range Domain*

Javier Ruiz-del-Solar[1,2], Rodrigo Verschae[1,2], and Daniel Kottow[1]

[1] Department of Electrical Engineering, Universidad de Chile
[2] Center for Web Research, Department of Computer Science, Universidad de Chile
{jruizd, rverscha}@ing.uchile.cl

Abstract. Most of the already proposed skin detection approaches are based on the same pixel-wise paradigm, in which each image pixel is individually analyzed. We think that this paradigm should be extended; context information should be incorporated in the skin detection process. Following this idea, in this article is proposed a robust and fast skin detection approach that uses spatial and temporal context. Spatial context implies that the decision about the class (skin or non-skin) of a given pixel considers information about the pixel's neighbors. Temporal context implies that skin detection is carried out considering not only pixel values from the current frame, but also taking into account past frames and general background reference information.

1 Introduction

Skin detection is a very popular and useful technique for detecting and tracking human-body parts. Its most attractive properties are: (i) high processing speed due to its low-level nature, and (ii) invariance against rotations, partial occlusions and pose changes. However, standard skin detection techniques are not robust enough. Changing lighting conditions and complex backgrounds containing surfaces and objects with skin-like colors are major problems in practical real-world applications.

For solving the mentioned drawbacks, many groups have centered their research on selecting the color-space most suitable for skin detection. Many different color models have been employed, among them: RGB, normalized RGB, HIS-HSV, YCbCr, YIQ, YES, YUV, CIE XYZ, CIE LUV, and Lab (see [11] for references). We believe that just selecting the "best" color space does not solve the mentioned drawbacks [5][1]. Some authors have used statistical models for solving the skin/non-skin classification problem (Mixture of Gaussians (MoG) [8][3] and histogram models [3][2]). In [9] a model that adapts a MoG to the image contents is used. In [6] an approach for skin detection under time-varying illumination using adaptive color histograms that model the color distribution over time is proposed. We think that statistical models are in the right direction; however they miss the benefits of using context information.

All mentioned approaches are based on the same pixel-wise paradigm, in which each image pixel is individually analyzed. We think that this paradigm should be

* This research was funded by Millenium Nucleus Center for Web Research, Grant P04-067-F, Chile.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 961 – 969, 2005.

extended incorporating context information to the skin detection process. Human *beings* can detect skin in real scenes, pictures and videos without problems. However, for a human being the classification of a single pixel as skin or non-skin is a very difficult task, because human skin detection is not a simple low-level process, but a process in which high-level mechanisms are also involved. If we think on the human perception of a blue ball under variable illumination, we will agree in that the ball is perceived blue as a whole, and not as a ball having blue patches and some other color patches generated by differences in illumination producing highlights and shadows. For having this kind of perception not only low-level color processing mechanisms for blue pixels and patches detection are involved, but also shape detection mechanisms for detecting the ball circular shape, and mechanisms for color constancy and interpolation [7]. In the same way, the detection of skin in a face or hand does not involve only low-level color processing mechanisms, but also high-level processes to assist the detection of skin (detection of hair, detection of clothes, etc), and some spatial diffusion mechanisms employed in any human segmentation process of colors and textures (cell mechanisms present in cortical area V4) [7].

Following these ideas we proposed a robust and fast skin detection approach that uses spatial context (spatial neighborhood information) and temporal context (temporal neighborhood and foreground pixels information). Spatial context implies that the decision about the class (skin/non-skin) of a given pixel considers information about the pixel's spatial neighbors. A diffusion process is implemented for determining the skin pixels. The aim of this process is not just the grouping of neighbor skin pixels, but the determination of skin areas where the color between neighbor pixels changes smoothly and at the same time the pixel *skinness*[1] keeps a minimal acceptable value. The seeds of the diffusion process are pixels with a high skinness value. Temporal context implies that skin detection is carried out considering the belonging of a pixel to the current background (BG) or foreground (FG) model. More explicitly, we model pixels belonging to the BG and pixels belonging to the FG by two sets of finite codebook vectors in the so-called spatial-range domain. In this domain pixels whose values are close enough to the BG or FG codebooks' vectors are classified as BG or FG, respectively. Given that we have codebooks "representing" (quantizing) the image pixels, we apply the spatial diffusion process only to the codebooks' vectors. These sets are much smaller than the number of image pixels in each frame and consequently we achieve a much faster processing time.

It is important to mention that the here-proposed work corresponds to an extension and integration of our algorithms: (i) *skindiff* [10], a skin detection algorithm for static images that uses local spatial context, and (ii) *tracey* [4], a BG and FG maintenance algorithm that works in the spatial-range domain.

2 The Proposed Skin Detection System

As already mentioned, the proposed skin detection systems works in the spatial range domain, incorporating spatial and temporal context information. Spatial context information for the skin detection is introduced using a diffusion process. Temporal

[1] We define Skinness as the belonging of a pixel to the skin class.

context is introduced by maintaining a BG/FG model, and by running the skin detection algorithm in: (1) the BG prototypes that were previously classified as skin, and (2) the FG prototypes. In this way the system is capable of dealing with static or slowly moving skin areas.

2.1 Vector Quantization of an Image on the Spatial-Range Domain

The spatial-range domain takes into account simultaneously the location and the intensity values of the image pixels. In this domain each image pixel has two parts, a spatial part (position) and a range part (intensity), where the range part may be written as a function of the spatial part [4]:

$$x_j = (x_j^s, x_j^r) = (x_j^s, I(x_j^s)) \quad (1)$$

The superscripts *s* and *r* denote the spatial (location) and range (value, intensity) parts of the pixels, respectively. By using this domain space an image can be *represented* (quantized or encoded) by a set of codebook vectors $C = \{c_i\}_{i=1..m}$ that cover all the images pixels. Each codebook vector will represent an equally-sized portion of the spatial-range domain, namely, all vectors lying inside a constant-sized hyper rectangle[1] $\bar{c}_i$ centered at the codebook vector:

$$x_j \in \bar{c}_i \Leftrightarrow \left\| x^s - c_i^s \right\| < \sigma_s \wedge \left\| x^r - c_i^r \right\| < \sigma_r$$

The size of the rectangle is given by σ_s and σ_r, which are constant and independent of *i*. They define the accuracy of the representation in the spatial and range sub domains. We say that a complete image *I* is represented accurately by a set of vectors C in the spatial range-domain iff:

$$\forall x_j \in I \, \exists c_i \in C (x_j \in \bar{c}_i) \quad (2)$$

Thus, using the algorithm shown in Figure 1 a given image can be learned, i.e. represented by a set of codebook vectors. Figure 2 shows an example of representation using the Lena image.

2.2 Tracey: The Background Model

Our BG model represents the BG by a set of codebook vectors in the spatial-range domain; image pixels not represented by this set will be output as the segmented FG To cope with the intensity variations of the BG, this set of codevectors is continuously updated. Our BG maintenance model consists of the set of BG codebook vectors, another set of codebook vectors representing the FG, and mechanisms that update the BG and the FG set of codebook vectors by removing old codevectors and adding new ones in order to obtain an accurate FG segmentation. The model needs to keep information about the current and recent past FG, because static objects in the FG should become BG after a reasonable time.

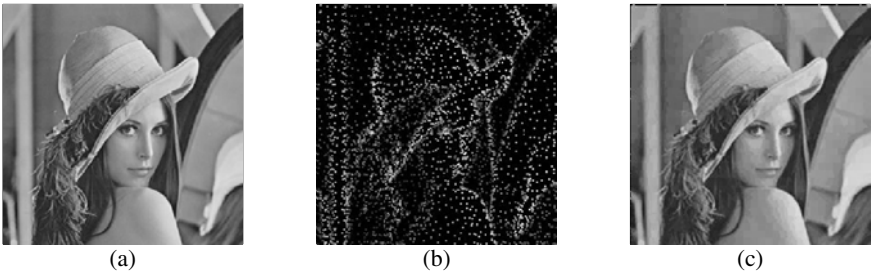
We use two sets of codebook vectors, one for the BG information and the other for the FG. Given an image pixel *x* and the codebooks C_{bg} (background) and C_{fg} (foreground) we define *DualLearn* (Fig. 3), which tells us if and which codebook represents the image pixel, and if none does, it adds the pixel to the FG. With the purpose of managing computational time effectively, we define a seed-growing

[1] In this work the L1 distance was used (for having a faster processing).

strategy for processing a complete image frame based on *DualLearn*. Given a set of image pixels *S*, *DualGrow* (Fig. 4) detects and processes areas where novel image content is found, while skipping areas that remain unchanged. Maintaining an accurate BG set of codebook vectors for an image sequence requires a method for removing codevectors when changes in the BG occur. We introduce a dynamical attribute s_i called score for every codevector c_i. If the codevector represents an image pixel at the codevectors spatial location, the score increases, otherwise it decreases (Figs. 5 and 6). This corresponds to a simple diffusion process using the outcome of the present representation ability as an exogenous input. Codevectors with a score below some threshold are considered obsolete and removed. On the other hand, a maximum score is used to identify static FG.

LearnImage*(I, C)*
foreach $x_j \in I$:
if $!\exists c_i \in C(x_j \in \bar{c}_i)$
$C \leftarrow C \cup \{x_j\}$

Fig. 1. Make *C* to represent an image *I*



(a) (b) (c)

Fig. 2. (a) Picture of Lena. (b) Set of codevectors that represents (a). (c) illustrates the quality of the codevectors by replacing each pixel value by the grey scale value of the codevector that represents (a).

2.3 Skindiff: The Skin Detection Algorithm

Skindiff works in two steps: (i) pixel-wise classification, and (ii) spatial diffusion (see fig. 7). The pixel-wise classification calculates the pixel skinness using a MoG model. The spatial diffusion refines the pixel-wise skin detection result, by taking into consideration neighborhood information for determining smooth skin segments, in which at least one pixel with a large skinness is present (see description in [10]).

One could think that the use of a diffusion process for video skin detection is slow; however this is not the case. Thanks to the use of LUTs (look-up-tables) for implementing the pixel-wise classification and a stack for avoiding a slow recursion in the diffusion process, plus the application of the diffusion algorithm in the codebook vectors (when processing video sequences), a very fast processing is obtained. Depending on its parameterization, the algorithm runs at 15-50 frames per second on a 1GHz Pentium 3 processor.

DualLearn(x, C_{bg}, C_{fg}):

if $\exists c_{i1} \in C_{bg}\,(x \in \bar{c}_{i1})$

return c_{i1}

else if $\exists c_{i2} \in C_{fg}\,(x \in \bar{c}_{i2})$

return c_{i2}

else

$C_{fg} \leftarrow C_{fg} \cup \{x\}$

Fig. 3. Make C_{fg} represent x if C_{bg} does not

DualGrow (I, C_{bg}, C_{fg}):

$S \leftarrow SampleSpace(I)$

while $S \neq \phi$

$x \leftarrow PopItem(S)$

$y \leftarrow DualLearn(x, C_{bg}, C_{fg})$

if $x == y$: // x was added to the foreground

foreach $x_n^s \in ConnectedNeighbors(x^s)$:

$PushItem(S, (x_n^s, I(x_n^s)))$

Fig. 4. Make C_{fg} *and* C_{bg} represent S and connected regions. *ConnectedNeighbors*(x^s) *returns the neighbors of* x^s in the two-dimensional lattice space of the image

UpdateForeground(I, C_{bg}, C_{fg}):

foreach $c_i \in C_{fg}$:

$x \leftarrow (c_i, I(c_i^s))$

if $x \in \bar{c}_i$

$s_i \leftarrow s_i(1-\tau)+\tau$

if $s_i \geq s_{static}$

$C_{bg} \leftarrow C_{bg} \cup \{c_i\}$

$C_{fg} \leftarrow C_{fg} \setminus \{c_i\}$

else

$C_{fg} \leftarrow C_{fg} \setminus \{c_i\}$

Fig. 5. Check for static objects in the foreground

UpdateBackground (I, C_{bg}):

foreach $c_i \in C_{bg}$:

$x \leftarrow (c_i, I(c_i^s))$

if $x \in \bar{c}_i$

$s_i \leftarrow s_i(1-\tau)+\tau$

else

$s_i \leftarrow s_i(1-\tau)-\tau$

if $s_i \leq s_{death}$

$C_{bg} \leftarrow C_{bg} \setminus \{c_i\}$

Fig. 6. Check for removal of background

2.4 Detecting Skin in Videos

All modules involved in the system are integrated in the *RecallImage* algorithm, which is computed at each frame, maintaining the BG/FG models and detecting the skin areas. *RecallImage* has the following processing step (see fig. 8):

A.Initialization: Before starting to process the frames, all sets are initialized. For the first frame, which is considered to be FG, a detailed representation is obtained. The first frame could also be considered as BG, in which case *skindiff* should be run on this set before start calling *RecallImage*, otherwise skin areas appearing on the first frame will be wrongly classified.

B.RecallImage:

1. The frame model is updated: let BG & FG vectors "represent" the image pixels).

2. Prototypes to be used in the diffusion are selected and added to S_{skin}. They

correspond to all prototypes of the FG and prototypes of the BG that were previously classified as skin. It is important to add prototypes of the BG to S_{skin}, otherwise static or slowly moving skin areas would be classified as non-skin and the diffusion would not work, because skin areas are usually homogenous and the diffusion needs "connected" areas to work. In fig. 9(c), FG pixels are sparse, and running the diffusion only on this set would give bad results. However when we consider also the BG codevectors previously classified as skin we obtain good results (see fig. 9(d)).

3. S*kindiff* is applied to the set S_{skin}. C_{skin}, contains the detected skin pixels.

4. Corresponding FG prototypes that were classified as skin are labeled. It is important to notice that a prototype can get the label *Skin* only if it is part of the FG. The label of the BG prototypes is never changed and a BG prototype will have a label *Skin* only if it got it when it was a FG prototype.

5. Static objects in the FG are learnt by the BG model.

6. Too old, non representative, BG vectors are discarded.

```
Skindiff (IM)
  {S_seed, S_min} ← find_seeds(IM)
  foreach ℑ ∈ S_seed
        PushItem(S_skin, ℑ )
        PushItem(Stack, ℑ )
        while(Stack ≠ φ)
          s ← PopItem(Stack)
          foreach x_j^s ∈ ConnectedNeighbors(s^s)
          if x_j ∉ S_skin AND x_j ∈ S_MIN
                if |x_j^r − s^r|^2 < T_tdiff^2
                   PushItem(S_skin, x_j)
                   PushItem(Stack, x_j)
  return S_kin
```

```
find_seeds(IM) {
  foreach x_i ∈ IM
    if g(x_i^r) > T_seed
        PushItem(S_seed, x_i)
    else if g(x_i^r) > T_min
        PushItem(S_min, x_i)
  return {S_seed, S_min}
}
```

IM: Set of prototypes representing the image.
S_{skin}: Final set of skin pixels, output of the algorithm.
S_{min}: Set of pixels that may be skin, S_{seed}: Set of seed pixels.
Stack: Stack data structure.
PopItem(S) removes and returns the first element from the ordered set *S*.
PushItem(S,x) adds *x* to the end of *S*.
ConnectedNeighbors(*s*) returns the neighbors of *s* in the 2D lattice of the image.
g() corresponds to the MoG described on [3] and it is implemented using a LUT [10].

Fig. 7. Skin detection algorithm

3 Results and Analysis

For testing the performance of the proposed algorithm, video sequences captured in our lab or obtained from Internet were used. The selected videos are considered difficult to segment; they have either changing lighting conditions or complex

backgrounds containing surfaces or static and moving objects with skin-like colors. The dataset consist of a total 5882 frames. A ground-truth was generated for about 0.35% of these frames. We are working on generating a larger testing dataset to obtain a better characterization of our system. The more complete dataset and their ground truth will be made available for future studies. We think that the here presented results show the potential of our system. Figure 9 and 10 show results for the segmentation of one frame of the sequence A and one frame of the sequence B. Sequence A has changing lighting conditions and in the sequence B different people enters and leaves a room.

The performance of the system and the effects of the different parameters are analyzed on hand of operation points defined by true positives (TP) and false positives (FP). TP are skin pixels correctly classified as skin and FP are non-skin pixels classified as skin. Figure 11 shows a cloud of operation points for the proposed algorithm for the sequences A and B. We are planning to perform a more exhaustive analysis considering all system parameters, for characterizing their effect on the detection rates, false positive rates, processing time and FG/BG segmentation.

Initialization: $S_{skin} \leftarrow \emptyset$, $C_{skin} \leftarrow \emptyset$, $C_{bg} \leftarrow \emptyset$, $C_{fg} \leftarrow \emptyset$

LearnImage(I, C_{fg}) // *or LearnImage*(I, C_{bg}); $C_{skin} \leftarrow$ skindiff(C_{bg})

RecallImage: // *process image frame* I

1. *DualGrow*(I, C_{bg}, C_{fg}) //*The model is updated to represent the frame*
2. $S_{skin} \leftarrow \emptyset$, $C_{skin} \leftarrow \emptyset$ //*Vectors to be used in the diffusion are selected:*

   foreach $c_i \in C_{fg}$: $S_{skin} \leftarrow S_{skin} \cup \{c_i\}$

   foreach $c_i \in C_{bg} \mid c_i^{Label} == Skin$: $S_{skin} \leftarrow S_{skin} \cup \{c_i\}$
3. $C_{skin} \leftarrow$ skindiff(S_{skin})
4. *foreach* $a_i \in C_{fg}$:

   if $\exists c_i \in C_{skin} (a_i \in \bar{c}_i)$: $a_i^{Label} \leftarrow Skin$
5. *UpdateForeground* (I, C_{bg}, C_{fg})
6. *UpdateBackground* (I, C_{bg})

Fig. 8. Proposed video skin detection algorithm

(a) (b) (c) (d)

Fig. 9. (a): Frame 271 of the sequence A. (b): Ground-truth of (a). (c): Detected Foreground. (d) Detected skin (white: foreground skin, grey: background skin)

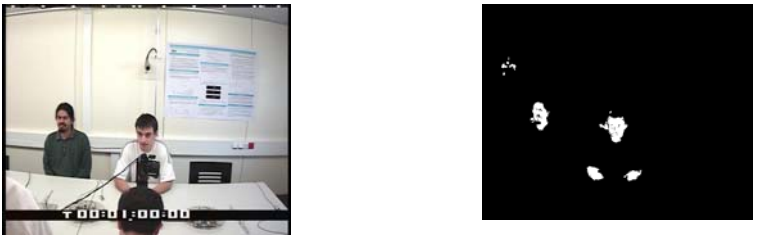


Fig. 10. Left: frame 11100 of sequence B. Right: detected skin in frame 11100

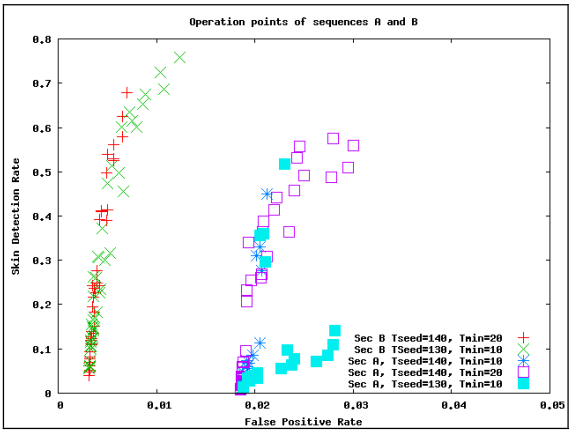


Fig. 11. Operation points of the proposed algorithm for 2 sequences

References

[1] A. Albiol, L. Torres, Ed. Delp, "Optimum Color Spaces for Skin Detection", *IEEE Int. Conf. on Image Proc. – ICIP 2001*, Greece, 2001.

[2] B. Jedynak, H. Zheng, and M. Daoudi, "Statistical Models for Skin Detection", IEEE Workshop Statistical Analysis in Computer Vision, together with CVPR 2003.

[3] M.J. Jones, and J.M. Rehg, "Statistical color models with application to skin detection", *Int. Journal of Computer Vision* 46(1): 81-96, 2002.

[4] D. Kottow, M. Köppen, and J. Ruiz-del-Solar, "A Background Maintenance Model in the Spatial-Range Domain", *2nd Workshop on Statistical Methods in Video* Processing (ECCV 2004 associated workshop), Prague, Czech Republic, May 16, 2004.

[5] M. Shin, K. Chang, and L. Tsap, "Does colorspace transformation make any difference on skin detection?", *Proc. IEEE Workshop on Appl. of Computer Vision*, Florida, USA, 2002.

[6] L. Sigal, S. Sclaroff, and V. Athisos, "Skin color-based video segmentation under time-varying illumination",*IEEE Trans. on Pattern Anal. and Machine Int.*,26(7):862-877,2004.

[7] L. Spillman and J. Werner (Eds.), *Visual Perception: The Neurophysiological Foundations*, Academic Press, 1990.

[8] M. H. Yang, and N. Ahuja, "Detecting human faces in color images", *Proc. IEEE Int. Conf. on Image Processing*, Chicago, Illinois, USA, 1: 127-130, 1998.

[9] Q. Zhu, K.-T. Cheng, C.-T. Wu, and Y.-L. Wu, "Adaptive Learning of an Accurate Skin-Color Model", *Proc. 6th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Seoul, Korea, 37-42, 2004.

[10] J. Ruiz-del-Solar, J. and R. Verschae, "Robust Skin Detection using Neighborhood Information", *Int. Conf. on Image Processing*, October 24 – 27, Singapore, October 2004.

[11] B. Martinkauppi, *Face Color under Varying Illumination – Analysis and Applications*, Doctoral Thesis, University of Oulu, Finland, 2002.

UBIRIS: A Noisy Iris Image Database

Hugo Proença and Luís A. Alexandre

Dep. Informatics, Universidade da Beira Interior,
IT - Networks and Multimedia Group, Covilhã,
R. Marquês D'Ávila e Bolama, 6200-001, Covilhã, Portugal
{hugomcp, lfbaa}@di.ubi.pt

Abstract. This paper presents a new iris database that contains images with noise. This is in contrast with the existing databases, that are noise free. UBIRIS is a tool for the development of robust iris recognition algorithms for biometric proposes.

We present a detailed description of the many characteristics of UBIRIS and a comparison of several image segmentation approaches used in the current iris segmentation methods where it is evident their small tolerance to noisy images.

1 Introduction

The use of biometric systems has been increasingly encouraged by both government and private entities in order to replace or increase traditional security systems.

Iris is commonly recognized as one of the most reliable biometric measures [27]: it has a random morphogenesis and no genetic penetrance [22]. In 1987 L. Flom and A. Safir [9] studied the problem and concluded that iris morphology remains stable through all human life, as well estimated the probability for two similar irises on distinct persons at 1 in 10^{72} [27].

In this paper we present UBIRIS [25], a new public and free iris database for biometric proposes. This database has characteristics that clearly distinguish it from the existing ones: CASIA [12] and UPOL [6]. Its main purpose is the evaluation of robust iris identification methodologies. The existing iris databases are noise free: this can be used to teste and develop segmentation and recognition algorithms that are able to work with images captured under near perfect conditions.

The emerging needs for a safer access (buildings, weapons, restricted areas) requires non-invasive, passive techniques. As an example we can think of a building access where the user does not need to look through a small hole to get his iris recognized, but a camera will take a photo (or several) of his iris while he approaches the door. This type of use is much less invasive and will enable the dissemination of iris recognition systems to everyday applications. UBIRIS is a tool for the development of such methods since it exhibits several types of image noise.

We compare the accuracy of several segmentation methodologies to demonstrate their small tolerance to heterogeneous images characteristics, such as the ones exhibited in UBIRIS.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 970–977, 2005.

1.1 Related Work

On a general iris recognition system four different stages can be identified: image capture, iris segmentation, feature extraction and feature comparison. In this section we focus on the iris image segmentation.

Since 1987, when the first relevant methodology was presented by Flom and Safir [9], many distinct approaches have been proposed. In 1993, J. Daugman [4] presents one of the most relevant methodologies, constituting the basis of many functioning systems. On the segmentation stage, this author introduces an integrodifferential operator to find both the iris inner and outer borders. This operator remains actual and was proposed with some minor differences in 2004 by [23]. Wildes [26] proposes iris segmentation through a gradient based binary edge map construction followed by circular Hough transform. This methodology is the most widely used, being proposed with minor several variants in [3], [11], [14], [17], [19], [18] and [20].

[16] proposes one simple method based on thresholds and function maximization in order to obtain two ring parameters corresponding to iris inner and outer borders.

Authors from [7] propose one iris detection method based on priori pupil identification. The image is then transformed into polar coordinates and the iris outer border is identified as the largest horizontal edge resultant from Sobel filtering. This approach may fail in case of non-concentric iris and pupil, as well as for very dark iris textures.

Morphologic operators were applied by [21] to obtain iris borders. They detect the inner border by applying threshold, opening and image closing and the outer border with threshold, closing and opening sequence.

Based on the assumption that the image captured intensity values can be well represented by a mixture of three Gaussian distribution components, authors in [13] propose the use of Expectation Maximization [5] algorithm to estimate the respective distributions parameters. They expect that 'Dark', 'Intermediate' and 'Bright' distributions contain the pixels corresponding to the pupil, iris and reflections areas.

Many of described approaches present a major disadvantage: the use of thresholds, usually to construct binary edge maps. This can be considered as a weak point regarding their robustness on image intensity changes.

2 UBIRIS

Despite the fact that many of the iris recognition approaches obtain almost optimal results, they do it under particularly favorable conditions, having few image noise factors. These conditions are not easy to obtain and require a high degree of collaboration from the subject, subjecting him to slower and uncomfortable image capture processes.

The aim of UBIRIS is related with this point: it provides images with different types of noise, simulating image captured without or with minimal collaboration from the subjects, pretending to become an effective resource for the evaluation and development of robust iris identification methodologies.

UBIRIS [25] database is composed of 1877 images collected from 241 persons during September, 2004 in two distinct sessions. It constitutes the world´s largest public and free available iris database at present date.

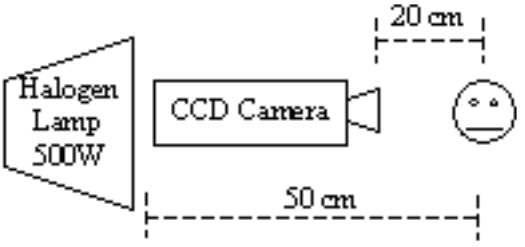


Fig. 1. Image capture framework

We used a Nikon E5700 camera with software version E5700v1.0, 71mm focal length, 4.2 F-Number, 1/30 sec. exposure time, RGB color representation and ISO-200 ISO speed. Images dimensions were 2560x1704 pixels with 300 dpi horizontal and vertical resolution and 24 bit depth. They were saved in JPEG format with lossless compression.

For the first image capture session, the enrollment, we tried to minimize noise factors, specially those relative to reflections, luminosity and contrast, having installed the framework represented in figure 1 inside a dark room.

In the second session we changed the capture location in order to introduce natural luminosity factor. This enabled the appearance of heterogeneous images with respect to reflections, contrast, luminosity and focus problems. Images collected at this stage pretend to simulate the ones captured by a vision system without or with minimal active collaboration from the subjects.

These noisy images will be compared to the ones collected during the enrollment stage.

All images from both sessions are classified with respect to three parameters ('Focus', 'Reflections' and 'Visible Iris') in a three value scale ('Good', 'Average' and 'Bad'). This classification was obtained manually and the results were: focus ($Good = 73.83\%, Average = 17.53\%, Bad = 8.63\%$), reflections ($Good = 58.87\%, Average = 36.78\%, Bad = 4.34\%$) and visible iris ($Good = 36.73\%, Average = 47.83\%, Bad = 15.44\%$)

2.1 PreProcessing

Preprocessing stage was essential in order to make possible the diffusion of UBIRIS on internet. We reduced the image size to $(width, height)$=$(400, 300)$, converted it to grayscale and saved in JPEG format with lossless compression. This process allowed us to obtain images with 27 KBytes average size and still enough quality for the execution of the algorithms.

3 Segmentation Methodologies

There are a large number of iris recognition methodologies that present almost optimal results, but only for well-segmented images. The fact that correspondent segmentation methods present much higher error rates, usually above 15%, is overlooked.

In this section we describe four iris image segmentation methods that are used in the experiments.

Daugman's method. The author [4] assumes both pupil and iris with circular form and applies an integrodifferential operator defined in equation (1).

$$max_{r,x_0,y_0}|G_\sigma(r) * \frac{\delta}{\delta r}\oint_{r,x_0,y_0}\frac{I(x,y)}{2\Pi r}ds| \quad (1)$$

This operator searches over the image domain (x, y) for the maximum in the blurred partial derivative with respect to increasing radius r, of the normalized contour integral of $I(x, y)$ along a circular arc ds of radius r and center coordinates (x_0, y_0) [4]. In pratical terms, this method searches in $\mathbb{N}^3$ for the circumference centre and radius that have the highest derivative value comparing to neighbour radius. This method, as showed in section 4.2, proved to be very effective on images with high intensity separability between iris, pupil and sclera regions. This method was chosen because, although it was proposed about 11 years ago, it remains as one of the most referred on iris segmentation literature and was the first methodology effectively implemented in a working biometric system [8].

Wildes method. Proposed in 1997, this methodology performs iris contour fitting in two steps [26].

First, the image intensity information is converted into a binary edge map. Second, the edge points vote for particular contour parameter values.

The first step is performed via gradient based edge detection [24] [1]. However, before this, the author proposes an histogram based approach to avoid problems with local minima that the active contour model's gradient might experience. Having this, in order to incorporate directional tuning, the image intensity derivatives are weighted to favor ranges of orientation. For example, on the iris/sclera border process, the derivatives are weighted to be selective for vertical edges.

The second step is made through the well known circular Hough transform [10]. This methodology is clearly the most common on iris segmentation approaches, having as principal disadvantage the dependence of threshold values for the edge maps construction. This fact can obviously constitute one weak point as we are concerned with robustness, which includes the ability to deal with heterogeneous image contrast and intensities.

Masek's method. Based on the methodology suggested on [26], this author proposes a method [20] that begins with the binary edge image map construction, using the Kovesi [15] edge detector, a variation of the well known Canny [2] edge detector.

The next step consists in applying the circular Hough transform in order to determine the iris/sclera border and then the one correspondent to iris/pupil. This methodology was included on the analysis essentially to represent several approaches on iris segmentation literature. In fact, several other authors propose minor variants to the Wildes [26] method, essentially to adjust the process to different image intensities and contrasts.

Liam and Chekima's method. This iris segmentation approach [16], is based on the fact that the pupil is typically darker than the iris and the iris darker than the sclera.

Based on this assumption, these authors propose the use of a thresholding technique that converts the initial captured grayscale image to binary. The threshold must be exactly calculated in order to join the pupil and the iris together in a dark region. Assuming that both components have circular form (iris and pupil), the next step consists on creating a ring mask that will run through the whole image searching for the iris/sclera border. The mask radius r and centre coordinates (x, y) will be defined by maximizing equation (2).

$$S = \sum_{\theta=0}^{2\Pi} (x + rcos(\theta), y + rsin(\theta)) \tag{2}$$

The next step consists in eliminating all image information outside the iris ring, and upgrading the threshold value in order to capture intensity dissimilarities between the iris and the pupil. Pupil/iris border determination is made according to the same methodology described for the iris/sclera border.
As we can see, this method's accuracy is strongly dependent of threshold values that have to be chosen by the user according to captured image characteristics.

4 Experiments

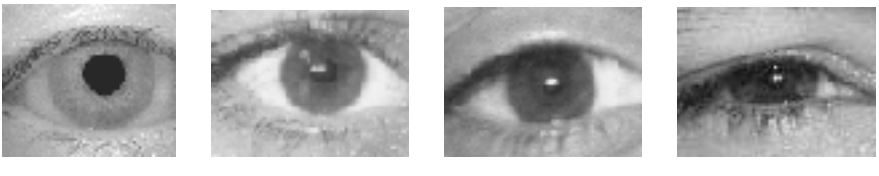
In this section we compare results obtained by the methodologies described in section 3 against UBIRIS and CASIA databases.

4.1 Databases

The reason for evaluating these methodologies against only the CASIA and UBIRIS databases is related with the fact that UPOL database just includes images from the internal part of the eye, having the segmentation work almost done. Figure 2 shows examples of CASIA database image and the different types of noise that are found on UBIRIS (reflections (2b), focus (2c) and small visible iris part (2d) noise).

4.2 Results and Discussion

On table 1 we show the results obtained by each described iris image segmentation method.



(a) CASIA image database. *(b) UBIRIS image database (Reflection).* *(c) UBIRIS image database (Focus).* *(d) UBIRIS image database (Visible Iris).*

Fig. 2. Image examples from two databases used in the experiments

Table 1. Segmentation accuracy results

Methodology	Parameters	UBIRIS	CASIA	Notes
Daugman	-	93.53%	54.44%	Method not dependent from any user value choice
Wildes	Hysteresis Thresholds: Hi=50, Low=44, Gaussian Kernel Dimension=5	89.12%	84.27%	Parameters optimized for UBIRIS database
Wildes	Hysteresis Thresholds: Hi=44, Low=39, Gaussian Kernel Dimension=5	81.28%	86.49%	Parameters optimized for CASIA database
Masek	Gaussian Kernel Dimension=5, Kovesi Parameters=(40,35)	87.12 %	81.85%	Parameters optimized for UBIRIS database
Masek	Gaussian Kernel Dimension=5, Kovesi Parameters=(39,34)	84.16 %	83.92%	Parameters optimized for CASIA database
Liam and Chekima	Threshold: 140	47.90%	56.33%	Parameters optimized for UBIRIS database
Liam and Chekima	Threshold: 150	41.66%	64.64%	Parameters optimized for CASIA database

All evaluated methods presented distinct accuracy levels on each database. This fact indicates that their accuracy is clearly dependent of the image characteristics, adding one relevant restriction to respective efficiency.

Daugman's [4] methodology, despite not being based on thresholds, presented the most heterogeneous results, having the best and the worst accuracy respectively on UBIRIS and CASIA databases. This fact can be easily explained by the lower contrast between iris and sclera eye parts on CASIA images. Trying to identify the circumference parameters (centre and radius) where the average intensity values have maximum derivative in respect to neighbour ones tends to identify circumferences tangent to pupil region. UBIRIS images have greater contrast between iris, pupil and sclera parts, thus yielding a better accuracy.

Approaches proposed by Wildes [26] and Masek [20] are similar on their methodology, therefore on their results, and have presented a more robustness behavior. However both methods are based on thresholds for constructing binary edges maps. This is an obvious disadvantage comparing to other image characteristics.

Apart from being the less accurate, thus obtaining worst results, methodology proposed by Liam and Chekima[16] was the less tolerant to image characteristic changes. This fact can be easily explained by the important role of the threshold operator that is the basis for both inner and outer border iris detection. In particular, probably motivated by the UBIRIS images characteristics, this methodology didn't at any circumstance reach the 50% accuracy, as opposite to CASIA database where accuracy was beyond 64%.

5 Conclusions

We presented a new public and free iris database for biometric proposes and described the most important aspects that distinguish it from the existing ones.

We encourage the use of this database by anyone who works or has interest on the area. It is available on `http://iris.di.ubi.pt` and can be downloaded with noise classification statistics.

We stress the minor importance that some authors give to segmentation stage, assuming that it is a trivial and error-free stage. Section 4.2 clearly showed that the selected methods deteriorate their performance in direct proportion with changes in image quality and characteristics.

Present work concerns the exhaustive test of existing iris image segmentation methodologies against available databases trying to identify and propose a more robust one.

Acknowledgements

We would like to thank all the persons that gave their contribute on UBIRIS images capture. In particular, we had the support from the 'Optics Center' and 'Multimedia Center' from Universidade da Beira Interior.

References

1. D. H. Ballard and C. M. Brown. *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, E.U.A., 1982.
2. J. Canny. A computational approach to edge detection. In *IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 8*, pages 679–698.
3. Jiali Cui, Yunhong Wang, Tieniu Tan, Li Ma, and Zhenan Sun. A fast and robust iris localization method based on texture segmentation. 2004. `http://nlpr-web.ia.ac.cn/english/irds/papers/cuijl/SPIE.pdf`
4. John G. Daugman. High confidence visual recognition of persons by a teste of statistical independence. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 11*, pages 1148–1161, U.S.A., 1993.
5. A. P. Dempster, N. Laird, and D. Rubin. Maximum likelyhood from incomplete data via the em algorithm. In *Journal of the Royal Statistic Society, vol. 39*, pages 1–38, 1977.
6. Michal Dobes and Libor Machala. Upol iris image database. 2004. `http://phoenix.inf.upol.cz/iris/`
7. Yngzi Du, Robert Ives, Delores Etter, Thad Welch, and Chein Chang. A new approach to iris pattern recognition. In *SPIE European Symposium on Optics/Photonics in Defence and Security*, 2004.
8. Jacqueline Emigh. The eyes have it. 2003. `http://securitysolutions.com/mag/security_eyes/`
9. Leonard Floom and Aran Safir. Iris recognition system, u.s. patent no. 4.641.349. 1987.
10. P. V. C. Hough. Method and means for recognizing complex patterns. 1962. U.S. Patent 3 069 654.
11. Junzhou Huang, Yunhong Wang, Tieniu Tan, and Jiali Cui. A new iris segmentation method for recognition. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR04)*, 2004.
12. Chinese Academy of Sciences Institute of Automation. Casia iris image database. 2004. `http://www.sinobiometrics.com`

13. Jaemin Kim, Seongwon Cho, and Jinsu Choi. Iris recognition using wavelet features. In *Kluwer Academic Publishers, Journal of VLSI Signal Processing, no. 38*, pages 147–156, The Netherlands, 2004.
14. W. K Kong and D. Zhang. Accurate iris segmentation method based on novel reflection and eyelash detection model. In *Proceedings of 2001 International Symposium on Inteligent Multimedia, Video and Speech Processing*, Hong Kong, May 2001.
15. Peter Kovesi. Matlab functions for computer vision and image analysis. 2004. `http://cs.uwa.edu.au/~pk/Research/MatlabFns/`
16. Lye Liam, Ali Chekima, Liau Fan, and Jamal dargham. Iris recognition using self-organizing neural network. In *IEEE, 2002 Student Confderence on Research and Developing Systems*, pages 169–172, Malasya, 2002.
17. Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang. Personal identification based on iris texture analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 12*, pages 1519–1533, U.S.A., 2003.
18. Li Ma, Yunhong Wang, and Tieniu Tan. Iris recognition based on multichannel gabor filtering. In *ACCV2002: The 5th Asian Conference on Computer Vision*, Australia, 2002.
19. Li Ma, Yunhong Wang, and Dexin Zhang. Efficient iris recognition by characterizing key local variations. In *IEEE Transactions on Image Processing, vol. 13, no. 6*, pages 739–750, U.S.A., 2004.
20. Libor Masek. Recognition of human iris patterns for biometric identification. 2003. `http://www.csse.uwa.edu.au/~pk/studentprojects/libor`
21. J. Mira and J. Mayer. Image feature extraction for application of biometric identification of iris - a morphological approach. In *IEEE Proceedings of the XVI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI03)*, 2003.
22. Alan Muron and Jaroslav Pospisil. The human iris structure and its usages. In *Acta Univ. Palacki, Phisica, vol. 39*, pages 87–95, 2000.
23. Ko Nishino and Shree K. Nayar. Eyes for relighting. *ACM Trans. Graph.*, 23(3):704–711, 2004.
24. W. K. Pratt. *Digital Image Processing*. Wiley, U.S.A., 1978.
25. Hugo Proença and Luís A. Alexandre. Ubiris iris image database. 2004. `http://iris.di.ubi.pt`
26. Richard P. Wildes. Iris recognition: an emerging biometric technology. In *Proceedings of the IEEE, vol. 85, no.9*, pages 1348–1363, U.S.A., 1997.
27. Gerald Williams. Iris recognition technology, iridian technologies, inc. 2001. `http://www.argus-solutions.com/pdfs/irisrecogwilliams.pdf`

Face Verification Advances Using Spatial Dimension Reduction Methods: 2DPCA & SVM

Licesio J. Rodríguez-Aragón, Cristina Conde,
Ángel Serrano, and Enrique Cabello

Universidad Rey Juan Carlos, c\ Tulipán, s/n,
E-28933, Móstoles, Madrid, Spain
{licesio.rodriguez.aragon, cristina.conde,
angel.serrano, enrique.cabello}@urjc.es
http://frav.escet.urjc.es

Abstract. Spatial dimension reduction called Two Dimensional PCA method has recently been presented. The application of this variation of traditional PCA considers images as 2D matrices instead of 1D vectors as other dimension reduction methods have been using. The application of these advances to verification techniques, using SVM as classification algorithm, is here shown. The simulation has been performed over a complete facial images database called FRAV2D that contains different sets of images to measure the improvements on several difficulties such as rotations, illumination problems, gestures or occlusion.
The new method endowed with a classification strategy of SVMs, seriously improves the results achieved by the traditional classification of PCA & SVM.

1 Introduction

Improving security and developing new smart environments are some of the key points in which biometry plays a most relevant role. Recent studies [1] have shown that technology is in very early stages of development to perform surveillance tasks at critical locations. However, simulations or real tests are crucial to obtain the required feedback in order to improve in the right direction.

Most of current face verification systems [2] relay not only in one algorithm but in an optimal ensemble of different methods that improve the global result. Classical dimension reduction methods, like Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA), and Gabor Filters [3], as well as other improved variations, like Independent Component Analysis (ICA) [4] and Kernel Principal Component Analysis (KPCA), are jointed and combined with classification algorithms, like Neural Networks and Support Vector Machines (SVM) [3,5]. The improvement of a single method implies the global improvement of the system.

Dimensional reduction methods applied to face verification tasks obtain a feature set for each image. Efficient feature extraction and selection schemes

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 978–985, 2005.

are crucial for a successful face verification system. Robustness in the feature extraction process to pose and light variations is aimed.

New advances on PCA method called Two-Dimensional PCA [6,7] have shortly been presented, and preliminar experiments and junctions of this new method with SVM are the focus of this work. Experiments are performed over a wide set of subjects, joined in a facial database of images, that combines different pose, gesture and illumination variations, which allow the measurement of the advances.

2 Feature Extraction

The PCA related feature extraction techniques require that 2D face images are transformed into a 1D row vector to then perform the dimension reduction [4]. The resulting image vectors belong to a high-dimensional image vector space where covariance matrices are evaluated with a high associated computational cost.

Recently, a Two-Dimensional PCA method (2DPCA) has been developed for bidimensional data feature extraction. 2DPCA is based on 2D matrices rather than 1D vectors, preserving spatial information.

2.1 Principal Component Analysis

Given a set of images $I_1, I_2, \ldots, I_N$ of height h and width w, PCA considers the images as 1D vectors in a $h \cdot w$ dimensional space. The facial images are projected onto the eigenspace spanned by the leading orthornormal eigenvectors, those of higher eigenvalue, from the sample covariance matrix of the training images. Once the set of vectors has been centered, the sample covariance matrix is calculated, resulting a matrix of dimension $h \cdot w \times h \cdot w$. It is widely known that if $N \ll h \cdot w$, there is no need to obtain the eigenvalue decomposition of this matrix, because only N eigenvectors will have a non zero associated eigenvalue [8]. The obtention of these eigenvectors only requires the decomposition of an $N \times N$ matrix, considering as variables the images, instead of the pixels, and therefore considering pixels as individuals.

Once the first d eigenvectors are selected and the proportion of the retained variance fixed, $\sum_1^d \lambda_i / \sum_1^N \lambda_i$, being $\lambda_1 > \lambda_2 > \cdots > \lambda_N$ the eigenvalues, a projection matrix A is formed with $h \cdot w$ rows and d columns, one for each eigenvector. Then a feature vector $Y_{d\times 1}$ is obtained as a projection of each image $I_{h\cdot w\times 1}$, considered as a 1D vector, onto the new eigenspace.

$$Y_{d\times 1} = A^T_{d\times h\cdot w} \cdot I_{h\cdot w\times 1} \tag{1}$$

2.2 Two-Dimensional Principal Component Analysis

The consideration of images $I_{h\times w}$ as 1D vectors instead as 2D structures is not the right approach to retain spatial information. Pixels are correlated to

their neighbors and the transformation of images into vectors produces a loss of information preserving the dimensionality. On the contrary, the main objective of these methods is the reduction of dimensionality and the least loss of information as possible.

The idea recently presented as a variation of traditional PCA, is to project an image $I_{h\times w}$ onto X by the following transformation [6,7],

$$Y_{h\times 1} = I_{h\times w} \cdot X_{w\times 1}. \tag{2}$$

As result, a h dimensional projected vector Y, known as projected feature vector of image I, is obtained. The total covariance matrix S_X over the set of projected feature vectors of training images $I_1, I_2, \ldots, I_N$ is considered. The mean of all the projected vectors, $\overline{Y} = \overline{I} \cdot X$, being $\overline{I}$ the mean image of the training set, is taken into account.

$$\begin{aligned} S_X &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N}(Y_i - \overline{Y})(Y_i - \overline{Y})^T \\ &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N}[(I_i - \overline{I})X][(I_i - \overline{I})X]^T \end{aligned} \tag{3}$$

The maximization of the total scatter of projections is chosen as the criterion to select the vector X. The total scatter of the projected samples is characterized by the trace of the covariance matrix of the projected feature vectors. Applying the criterion to (3) the following expression is obtained,

$$J(X) = tr(S_X) = X^T[\frac{1}{N}\sum_{i=1}^{N}(I_i - \overline{I})^T(I_i - \overline{I})]X. \tag{4}$$

What is known as image covariance matrix G defined as a $w \times w$ nonnegative matrix can be then directly evaluated using the training samples,

$$G = \frac{1}{N}\sum_{i=1}^{N}(I_i - \overline{I})^T(I_i - \overline{I})]. \tag{5}$$

The optimal projection axis X_{opt} is the unitary vector that maximizes (4), which corresponds to the eigenvector of G of largest associated eigenvalue.

Usually, as well as in traditional PCA, a proportion of retained variance is fixed, $\sum_1^d \lambda_i / \sum_1^w \lambda_i$, where $\lambda_1 > \lambda_2 > \cdots > \lambda_w$ are the eigenvalues and $X_1, X_2, \ldots, X_d$ are the eigenvectors corresponding to the d largest eigenvalues.

Once d is fixed, $X_1, X_2, \ldots, X_d$ are the orthornormal axes used to perform the feature extraction. Let $V = [Y_1, Y_2, \ldots, Y_d]$ and $U = [X_1, X_2, \ldots, X_d]$, then

$$V_{h\times d} = I_{h\times w} \cdot U_{w\times d}. \tag{6}$$

A set of projected vectors, $Y_1, Y_2, \ldots, Y_d$, are obtained and in 2DPCA each principal component is a vector instead of an scalar as in traditional PCA. A feature matrix $V_{h\times d}$ is produced containing the most discriminating features of image I.

2.3 Image Reconstruction

In both methods, PCA and 2DPCA, a reconstruction of the images from the features is possible. An approximation of the original image with the retained information determined by d is obtained.

$$\begin{array}{ll} \widetilde{I}_{h\cdot w\times 1} = A_{h\cdot w\times d}\cdot Y_{d\times 1} & \text{PCA image reconstruction.} \\ \widetilde{I}_{h\times w} = V_{h\times d}\cdot U^{T}_{d\times w} & \text{2DPCA image reconstruction.} \end{array} \tag{7}$$

3 Classification with SVM

SVM is a method of learning and separating binary classes [9], it is superior in classification performance and is a widely used technique in pattern recognition and especially in face verification tasks [5].

Given a set of features $y_1, y_2, \ldots, y_N$ where $y_i \in \mathbb{R}^n$, and each feature vector associated to a corresponding label $l_1, l_2, \ldots, l_N$ where $l_i \in \{-1, +1\}$, the aim of a SVM is to separate the class label of each feature vector by forming a hyperplane

$$(\omega \cdot y) + b = 0, \quad \omega \in \mathbb{R}^n, b \in \mathbb{R}. \tag{8}$$

The optimal separating hyperplane is determined by giving the largest margin of separation between different classes. This hyperplane is obtained through a minimization process subjected to certain constrains. Theoretical work has solved the existing difficulties of using SVM in practical application [10].

As SVM is a binary classifier, a *one vs. all* scheme is used. For each class, each subject, a binary classifier is generated with positive label associated to feature vectors that correspond to the class, and negative label associated to all the other classes.

3.1 Facial Verification Using SVM

In our experiments a group of images from every subject is selected as the training set and a disjoint group of images is selected as the test set. The training set is used in the feature extraction process through PCA and 2DPCA. Then, the training images are projected onto the new orthornormal axes and the feature vector (PCA), or vectors (2DPCA), are obtained. For each subject the required SVMs are trained.

For both methods, PCA and 2DPCA, the same amount of retained variance has been fixed, giving as result different values of d (number of considered principal components) for each reduction method. When training and classifying PCA features, each image generates one feature vector $Y_{d\times 1}$ and one SVM is trained for each subject, with its feature vectors labelled as $+1$ and all the other feature vectors as -1. On the other hand, for feature vectors obtained from 2DPCA, each image generates a set of projected vectors, $V_{h\times d} = [Y_1, Y_2, \ldots, Y_d]$, and consequently for each subject d SVMs need to be trained, one for each feature vector Y_i.

Once the SVMs are trained, images from the test set are projected onto the eigenspace obtained from the training set. The features of the test set are classified through the SVMs to measure the performance of the generated system. For the SVM obtained from the PCA feature vectors, the output is compared with the known label of every test image. However, for the ensemble of SVMs obtained from the 2DPCA feature vectors, the d outputs are combined through a weighted mean. Every output is weighted with the amount of variance explained by its dimension, that means that each output will be taken in account proportionally to the value of the eigenvalue associated to the corresponding eigenvector: $\lambda_i / \sum_{j=1}^{d} \lambda_j$ is the weight for the $i-$SVM, $i = 1, 2, \ldots, d$.

To measure the system performance a cross validation procedure is carried out. Results are then described by using Receiver Operating Curve, ROC curve, as there are four possible experiment outcomes: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The system threshold can then be adjusted to more or less sensitiveness, but in order to achieve fewer errors new and better methods, like 2DPCA, are required.

4 Images Database: FRAV2D

The Face Recognition and Artificial Vision[1] group (FRAV) at the Universidad Rey Juan Carlos, has collected a quite complete set of facial images for 109 subjects. All the images have been taken under controlled conditions of pose and illumination. 32 images were taken of each subject, being 12 frontal, 4 preforming a 15° rotation, 4 performing a 30° rotation, 4 with zenithal instead of diffuse illumination, 4 performing different gestures and 4 occluding parts of the face. A partial group of this database is freely available for research purposes.

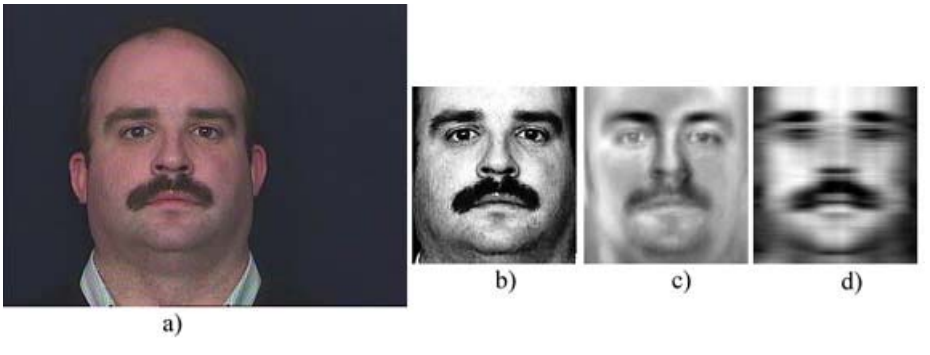


Fig. 1. a) One of the original frontal images in the FRAV2D database. b) Automatically selected window containing the facial expression of the subject in equalized gray scale. c) and d) Reconstructed images (7), for the 60% of retained variance, from PCA and 2DPCA projections respectively.

[1] http://frav.escet.urjc.es

The images are colored and of size 240×320 pixels with homogeneous background color. A window of size 140×130 pixels containing the most meaningful part of the face, has been automatically selected in every image and stored in equalized gray scale. That is the information that will be analyzed through the dimension reduction and classification methods (Fig. 1).

5 Design of Experiments

The purpose of the following experiments is to confront the performance of the traditional PCA method to the new proposed 2DPCA method in the task of face verification through SVM. The retained variance for both methods has been fixed up to 60% resulting values of $d = 20$ for PCA and $d = 5$ for 2DPCA.

Each experiment has been performed for 100 randomly chosen subjects from the whole FRAV2D. In all the experiments, the train set for the extraction of the feature vectors and for the classifiers training is formed by 8 frontal images of each subject. Then, the classifiers have been tested over 5 different groups of images. Firstly, the 4 remaining frontal images for each subject have been used to perform the cross validation process. In a second experiment, the 4 images obtained with zenithal illumination have formed the tests set. The 4 15° images have been selected to measure the performance of the system to rotations. In the fourth experiment 4 images with gestures have been used. And finally, the 4 occluded images for each subject have formed the test set.

Results for each experiment are presented as ROC curves, showing the compared performance of the verification process using PCA and 2DPCA. True positive rate (TP), that is the proportion of correct classifications to positive verification problems, and true negative rate (TN), that is the proportion of correct classifications to negative verification problems, are plotted (Fig. 2).

Besides, the equal error rate (EER), that is the value for which false positive rate (FP) is equal to false negative rate (FN), is presented for each experiment in Table 1.

Table 1. EER, values for which the rate of incorrect classification of positive verifications is equal to the rate of incorrect classification of negative verifications, for each dimension reduction method

Experiment	PCA	2DPCA
1) Frontal Images	1.8	1.0
2) Zenithal Illumination	6.6	2.1
3) 15° Rotated	27.3	18.3
4) Gesture Images	15.8	12.9
5) Occluded Images	34.7	29.5

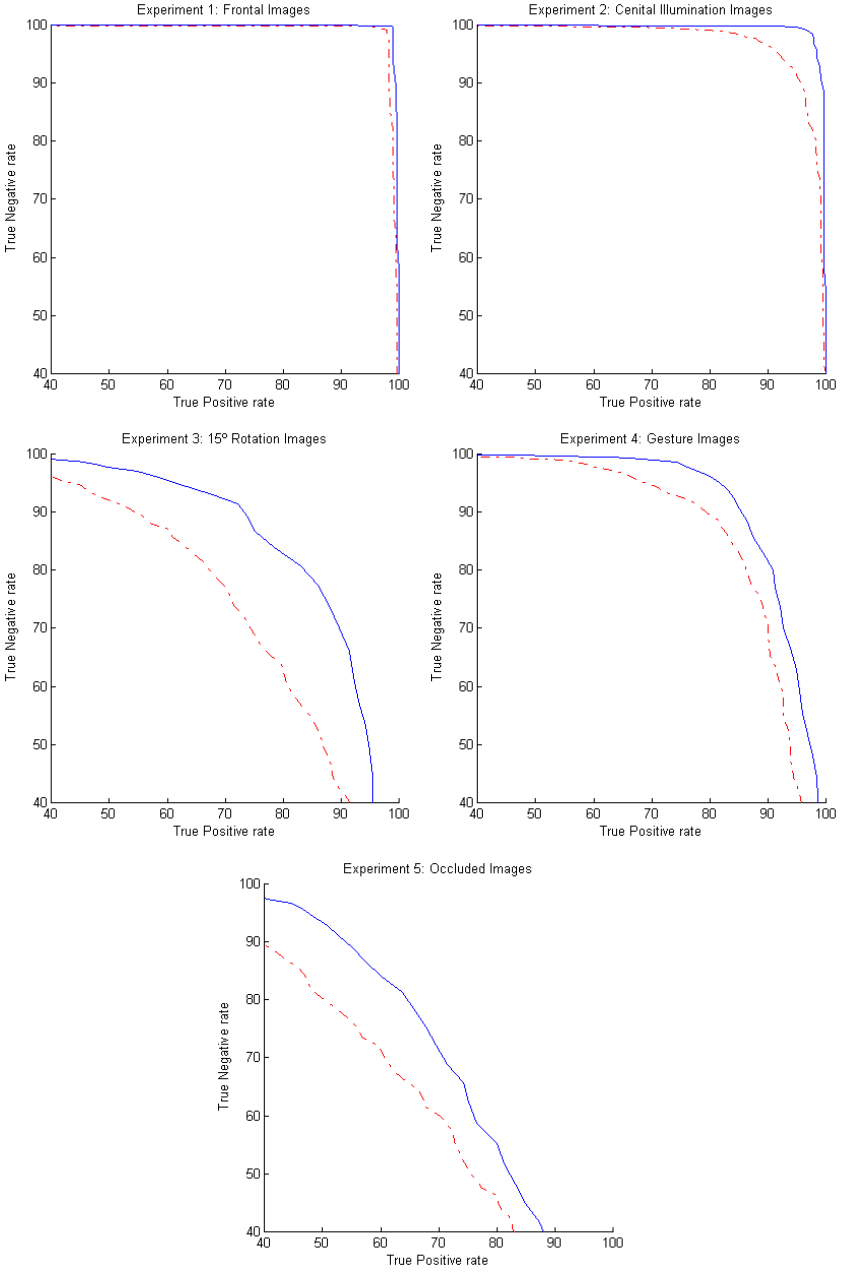


Fig. 2. ROC curve for each experiment, with TP rate in abscises and TN rate in ordinates. The performance of both classifying strategies PCA, dashed line, versus 2DPCA, solid line, is shown.

6 Conclusions

From the ROC curves (Fig. 2) and the EER values presented (Table 1) for each experiment and method, better results are achieved for the spatial reduction method 2DPCA. This improvement is evidently present in all the experiments

Problems merged from illumination and gestures are more accurately solved with this improvement of the dimension reduction and the classification strategy used. Serious improvements have also been done as far as rotated and occluded images are concerned. The good performance reached by traditional PCA in frontal images has even been improved.

The ensemble of the here proposed strategy with other methods and techniques used in face verification tasks will be an aim to be reached in the future developments.

Acknowledgments

Authors would like to thank César Morales García for his enthusiastic work. Also thanks must be given to every one that offered his help to join FRAV2D data base. This work has been partially supported by URJC grant GVC−2004 − 04.

References

1. Bowyer, K. W.: Face recognition technology: security versus privacy. IEEE Technology and society magazine. **Spring** (2004) 9–20
2. Messer, K., Kittler, J., Sadeghi, M., et al. : Face authentification test on the BANCA database. Proceedings of the International Conference on Pattern Recognition. (2004) 523–532.
3. Pang, S., Kim, D. and Bang, S. Y.: Memebership authentication in the dynamic group by face classification using SVM ensemble. Pattern Recognition Letters **24** (2003) 215–225.
4. Kim, T., Kim, H., Hwang, W. and Kittler, J.: Independent Component Analysis in a local facial residue space for face recognition. Pattern Recognition. **37** (2004) 1873–1885.
5. Fortuna, J. and Capson, D.: Improved support vector classification using PCA and ICA feature space modiffication. Pattern Recognition **37** (2004) 1117–1129
6. Yang, J. and Yang, J.: From image vector to matrix: a straightforward image projection technique–IMPCA vs. PCA. Pattern Recognition **35** (2002) 1997–1999.
7. Yang, J., Zhang, D., Frangi and F., Yang, J.: Two-Dimmensional PCA: A new approach to apperance-based face representation and recognition. IEEE Transacctions on Pattern Recognition and Machine Intelligence. **26** (2004) 131–137.
8. Turk, M. and Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neurosicience. **3** (1991) 71–86.
9. Cortes, C. and Vapnik, V.: Support vector network. Machine Learning. **20** (1995) 273–297.
10. Joachims, T.: Making large scale support vector machine learning practical. In: Advances in Kernel Methods: Support Vector Machines. MIT Press, Cambridge, MA.

Asymmetric 3D/2D Processing: A Novel Approach for Face Recognition

Daniel Riccio[1] and Jean-Luc Dugelay[2]

[1] Universitá di Salerno, via Ponte Don Melillo, 84084 Fisciano, Salerno, Italy
driccio@unisa.it
[2] Institut Eurecom, CMM, 2229 route des Cretes,
B.P. 193, F-06904, Sophia Antipolis, Cedex
Jean-Luc.Dugelay@eurecom.fr

Abstract. Facial image analysis is very useful in many applications such as video compression, talking heads, or biometrics. During the last few years, many algorithms have been proposed in particular for face recognition using classical 2-D images. Face is fairly easy to use and well accepted by people but generally not robust enough to be used in most practical security applications because too sensitive to variations in pose and illumination. One possibility to overcome this limitation is to work in 3-D instead of 2-D. But 3-D is costly and more difficult to manipulate and then ineffective to authenticate people in most contexts. Hence, to solve this problem, we propose a novel face recognition approach that is based on an asymmetric protocol: enrolment in 3-D but identification performed from 2-D images. So that, the goal is to make more robust face recognition while keeping the system practical. To make this 3-D/2-D approach possible, we introduce geometric invariants used in computer vision within the context of face recognition. We report preliminary experiments to evaluate robustness of invariants according to pose variations and to the accuracy of detection of facial feature points. Preliminary results obtained in terms of identification rate are encouraging.

1 Introduction

Biometric technologies that currently offer greater accuracy such as iris and fingerprint, require, however, much greater cooperation from the user and are too much invasive in some cases. Face Recognition includes a good compromise between people acceptance and reliability (in controlled environments). In the last years, many strategies have been proposed in order to solve the recognition problem, mainly addressing problems such as changes in expression, pose and illumination. Recent works attempt to solve the problem directly on a 3D model of the face. Indeed, a 3D model provides more geometrical information on the shape of the face and is unaffected by illumination and pose variation. The development of 3D acquisition systems and then the 3D capturing process are becoming cheaper and faster too. This definitely makes the 3D approach more and more applicable to real situations out of the laboratories. However, unlike 2D face recognition, there are yet few works on 3D range images.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 986–993, 2005.

In [3] Huang *et al.* develop a component-based recognition method based on a 3D morphable model. At first the 3D model is generated from two different views of the subject and then a number of synthetic views are rendered from the model changing pose and illumination. Yet the database consist of so few people (6 subjects) and training/testing images are generated from the same models. Bronstein *et al.* in [2] suggest the use of a canonical form, which consists of an isosurface of the face shape, with the flattened texture mapped on, and where principal component analysis is used to decompose the obtained canonical image. A method, that works on 2D, but using however a 3D morphable model for training/testing is shown in [1]. Despite of its performances in terms of recognition rate, the greatest drawback of this approach is its computational cost.

The most part of the proposed method applies only to the 3D range images, but even if the 3D acquisition is becoming cheaper, the problem of the sensitivity of the capturing process remains. This point out the usefulness of an approach, that profits by a 3D model based enrolment, but only needs of a 2D view of the model for testing. This is one of the main motivations for which a new framework for 3D/2D face recognition is introduced here. The proposed approach is based on 3D projective invariants, used for long time in computer vision, recognizing object with rigid surface. As the capacity of recognizing objects in a scene, regardless of their orientation, is an important goal in the computer vision from long time, several relevant papers have been published on this topic. They describe a lot of measures or ratio of distances that are invariant with respect to projective and/or perspective transformations. There are no geometric invariants for an unconstrained set of points in the space. However, interesting properties have been inferred when points in the 3D space are collinear, coplanar or their structure in the space is well described in a given way.

1.1 Geometric Invariants in Face Recognition

Given *inhomogeneous* coordinates $x = \left(x^1, x^2, \ldots, x^m\right)^t$, where $x \in R^m$, the correspondent *homogeneous* coordinates of the point are $z = \left(z^1, z^2, \ldots, z^m, z^{m+1}\right)^t$, where $x^l = z^l / z^{m+1},\ l = 1, \ldots, m,\ z^{m+1} \neq 0$. The homogeneous coordinates are a more general way to represent points, requiring the constraint that $\exists l \in \{1, \ldots, (m+1)\} \ni z^l \neq 0$. By means of this mapping, the projective transformation in R^m, can be easily managed as linear transformation in R^{m+1}. Thanks to this representation, the most of the ratios among distances in the space can be represented as ratio of determinants of the corresponding point coordinates.

The are two main categories of invariants. The first category, namely 2D image based invariants, does not require the 3-D object to be computed but constraints about the localization of feature points are important. On the contrary, the second category, namely 3D image based invariants, requires the 3-D object or at least 3 points of view but is very flexible about the repartition of the anchor points. Besides, to extract the invariants from the 3D model rather than from a given set of images is advantageous for two main reasons: 1) in some of the available images, the points to be selected could be occluded, while with

the 3D model, any point configuration can be considered; 2) On a set of views the points must be selected and their value is then affected by the localization error. On the contrary, on the 3D model a correct calculation of the invariants can be performed.

Given four collinear points $z_1, z_2, z_3, z_4 \in R^2$, the simplest invariant is their cross ratio that can be written as:

$$c(z_1, z_2, z_3, z_4) = \frac{M(1,3) \cdot M(2,4)}{M(1,4) \cdot M(2,3)} \text{ with } M(i,j) = \begin{vmatrix} x_i & x_j \\ 1 & 1 \end{vmatrix}. \quad (1)$$

This property can be extended to five points, which lie on the same plane $z_i \in R^3/(0,0,0), i = 1, \ldots, 5$, so that two functionally independent projective invariants hold:

$$c_1 = \frac{M(1,2,4) \cdot M(1,3,5)}{M(1,2,5) \cdot M(1,3,4)} \text{ and } c_2 = \frac{M(2,1,4) \cdot M(2,3,5)}{M(2,1,5) \cdot M(2,3,4)}. \quad (2)$$

At last Zhu *et al.* [6] demonstrated that given six point in a 3D space A, B, C, D, E and F, which lie on two adjacent planes, as shown in Fig. 1 (c), the cross-ratio of the areas of the corresponding triangles is a projective invariant, if no three of each set of four coplanar points are collinear:

$$I = \frac{P_{ABD} \cdot P_{FEC}}{P_{ABC} \cdot P_{FED}} \quad (3)$$

It is important to note that the goal of previous works using 3-D model based invariants was to discriminate objects that are rigid and obviously different.

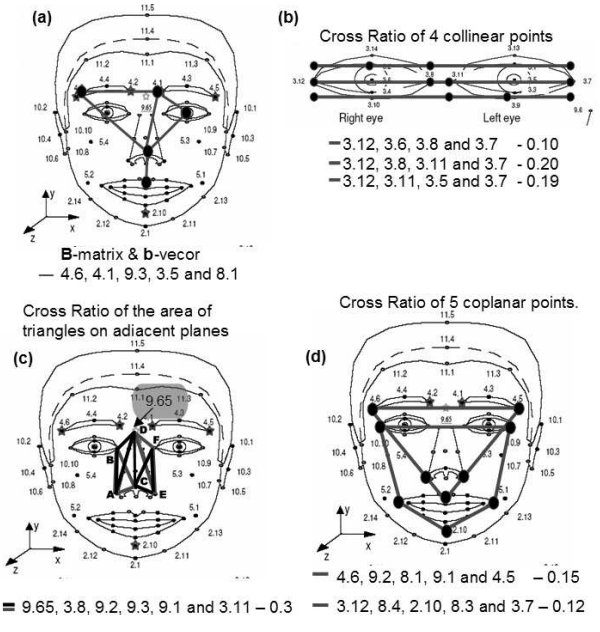


Fig. 1. Graphic representation of the control points and corresponding invariants

This is not the case for faces that are flexible and similar from one person to another one, so that even slight perturbations on the control points can result in a misclassification of the corresponding face. Therefore, a very important feature for a 3D model based invariant is then its sensitiveness to the noise on the control points. For this reason one of the more noise-insensitive invariant has been chosen. It has been proposed by Weinshall in [5].

Given an object in the 3D space, its representation is closely related to the reference frame. However choosing three points on the object, corresponding to three linear independent vector p_i, p_j, p_k, a new reference system can be defined, which make the representation of the object invariant to projective transformations. Every point $p_l \in R^3$ on the object can be written as a linear combination of this basis: $p_l = b_1^l p_i + b_2^l p_j + b_3^l p_k$. The vector $b^l = \left(b_1^l, b_2^l, b_3^l\right)$ represents an affine invariant. The Euclidean metric information on the basis points is represented by means of their inverse Gramian matrix $B = G^{-1}$.

In [5] the author also proposed two different kind of invariants, defined by means of a set of four/five non coplanar points and the inverse Gramian matrix B. Indeed, consider an object composed of four non-coplanar 3D points, where $\{P_l\}_{l=0}^3$ denote the 3D coordinates of the four points in some reference frame. Assume $P_0 = (0, 0, 0)$ without loss of generality and let $\{p_l\}_{l=1}^3$ denote the 3D vectors corresponding to the three remaining points. Given the image coordinates of the four points $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$, where $x_0 = 0, y_0 = 0$ is assumed. Let $x = (x_1, x_2, x_3)$ and $y = (y_1, y_2, y_3)$. The first rigid invariant is given by the equation 4, while retaining the same notation, the second invariant can be defined, combining the vectors of the basis $b = (b_1, b_2, b_3)$ and the image coordinates of five points, as shown by the equation 5.

$$f_B = \frac{\left|x^T B y\right| + \left|x^T B x - y^T B y\right|}{|x| \, \|B\| \, |y|} \tag{4}$$

$$f_b = \frac{\left|x_4 - \sum_{i=1}^3 b_i x_i\right|}{|x| \, |b|} + \frac{\left|y_4 - \sum_{i=1}^3 b_i y_i\right|}{|y| \, |b|}. \tag{5}$$

The value of the functions f_B and f_b is zero for all the views of the object, that the matrix B or the vector b describe. They are normalized by means of the norm of the vectors x, y and the matrix B or the vector b respectively, so that their value does not depend on the distance between the object and the camera.

2 The Proposed Approach

Works dealing with 3D invariants have been devoted to rigid objects, but face is flexible. This represents the first problem to be solved, when applying invariants in 3D face recognition. Indeed changes in expression modify the geometry of the face, more than anything else, jeopardizing the results. This point out the significance of choosing carefully the points, namely *control points*, used in next computations for the extraction of the invariants.

2.1 Feature Extraction

The 19 control points have been chosen as a subset of the Feature Points defined in MPEG-4 standard [4]. Fig. 1 highlights that the point 9.65 is not present in the MPEG-4 standard, but it has been inserted, so that two adjacent planes can be made up. Almost all the considered invariants impose some hypotheses on the configuration of the control points, such as collinearity or coplanarity. However the face is not a rigid surface and to find control points which both respect the required hypotheses and that are easy to locate, turns in a difficult task. Therefore it makes sense that the required conditions are tested for each candidate configuration, in order to assure a real good approximation of the theory. For the 2D image based invariants, that is all the ratios, the collinearity has been assessed computing the approximation error by means of a linear regression. In the same way, in order to test the coplanarity of each configuration of five points, the approximation error has been computed by means of a plane regression. Then the smaller the approximation error is, better the quality of the chosen configuration will be.

In Fig. 1 are reported the most part of the 2D image based invariants that have been used and the corresponding approximation error, computed on a large set of 3D face models. On the contrary, for the 3D model based invariants the only need is the non collinearity and/or non coplanarity of the control points, which is easy to achieve properly choosing each configuration. In order to optimize the choice of the control point configurations, opting for those providing the greatest discriminating power, the distribution of the control points with respect to their average position has been investigated. The models have been lined up with respect to the nose tip and the centroid has been drawn out for each of the 19 class of controls points, further the standard deviation of each class has been calculated with respect to its centroid. According to these results, for the f_B and f_b invariants eight configurations of four and five points have been chosen respectively.

2.2 Classification Process: Enrolment/Testing

When a new user has to be enrolled, the system acquires both the 3D shape and the 2D texture of his/her face. The control points are then located on the 2D texture of the face, while the corresponding 3D points are automatically retrieved on the 3D shape. All the 2D image based invariants (all the ratios) are computed by the (x, y) coordinates of the control points. They consist in scalar values (ratio of distances), so they can be stored in the first part of the feature vector. On the contrary, for each 3D model based invariant f_B the B matrix is computed and its $B_{i,j}$ items are then inserted in the second part of the feature vector. At last, the b vector is computed for each of the f_b invariant and the corresponding b_i values are stored in the last part of the feature vector.

The testing process is partitioned in two steps, in order to make this task efficient as well as effective. Let be F a query image submitted to the system. First of all, the 19 control points are located on F. Some of them are used to

compute all the cross ratios, as described in Section 1.1, forming the first part of the feature vector V_F for the face F. This vector is then used to query the system in order to retrieve a subset of only K of the N subjects in the database, which have to be further authenticated by the f_B and f_b invariants, according to a voting strategy, that is for each of the eight configuration of the control points, the corresponding f_B and f_b invariants votes for one of the K retrieved subjects, and that one receiving the most of votes is returned as the correct identity.

In other words, the proposed method performs in two sequential steps. The former is a pruning operation on the database, resulting a subset of the face database that retains best candidates, while the latter consists in the real identification task, in which the retrieved subjects are identified by means of the 3D model based invariants. In general, to reduce the number of the subjects to be identified allows a noticeable drop in the computational cost. Indeed, in this case the feature vectors are organized in a structured manner, such as a tree, then a subset of K good candidates can be retrieved by the screening operation in time $O(logN)$ and identified by the 3D model based invariants in time $O(K)$, intead of $O(N)$ of the full identification.

3 Experimental Results

Since the invariants are calculated only from the control points, which are detected on the image by hand at the moment, they are naturally robust against the illumination variations. The main problem that is faced in the experiments is therefore the sensitiveness of the algorithm with respect to the pose variations and inaccuracy of detection of the control points. The proposed method has been tested on a property database realized by Eurecom.

All the faces have been acquired by means of the Geometrix system [7], which uses two cameras (up and down), in order to extract the 3D shape of the face. The database consists of 50 people acquired in normal conditions of expression, pose and illumination. The age of the subjects ranges between 20 and 50 years, 40 of them are male and 10 females, further the most are Caucasians. Three models for each subject have been considered.

In the first experiment the goal is to investigate how the discriminant power of the 2D image based invariants drops respect to the parameter K and the pose changes. The database has been divided in two subsets, a probe and a gallery set, respectively. For each subject, one model has been inserted in the probe set and the remaining two in the gallery. In this way a manual localization of the control points is properly simulated. Furthermore K represents a tuning parameter for the system, ranging between 1 and N. In this case $K \in [1, 100]$, two models for each of the 50 subjects. Notice that all the models in the gallery are considered distinct. The results of this experiment are reported in Fig. 2 (a).

In the second experiment 50 models have been considered, in order to assess the performances of the system respect to the accuracy of locating the control points. The models in the probe and gallery set are the same, so there is no initial error on the control points, while K is fixed to 20. Five different pose

have been considered, while increasing noise is added to the coordinates of the control points. The noise is generated randomly in the range $[-2e_{rr}, 2e_{rr}]$ with mean $e_{rr} = 0.5, 1, ..., 5$ (results for $e_{rr} = 0$ has not been reported because they are all ones). In Particular, e_{rr} represents the pixel accuracy of locating the control points, with respect to a 256×256 image. The results are shown in Fig. 2 (b). The precision of the control point localization has been also drawn out on the models of the real database, measuring the mean error between the models in the probe and the corresponding ones in the gallery. An error of about 3.2 pixels has been estimated. Indeed, results marked with the ellipses confirm that when feature points are manually selected (ellipse in Fig. 2 (a)), the average error in localization is somewhat equivalent to a additional noise of 3 pixels in exact but artificial conditions (ellipse in Fig. 2 (b)).

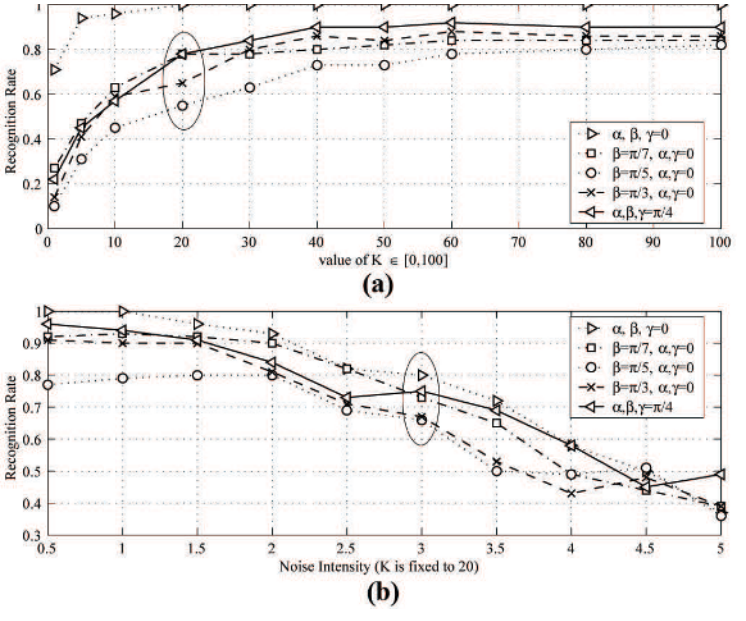


Fig. 2. (a) Recognition rate of the system, varing pose and the K parameter. (b) The probability that the correct subject is the first answer or is in the first 5 answers, when increasing random noise is added and $K = 20$.

The results of the first experiment suggest that the value of K must be proportional to the magnitude of pose variation, which can be estimated from the distribution of the control points. Therefore it makes sense that gometric invariants could play a role in a multimodal face recognition system, which also takes into account for the information provided by the texture image. At last both the experiments underline that 3D model based invariants are more powerful of the 2D image based geometric invariants, taking an interest in a more comprehensive study in this sense.

4 Conclusion and Remarks

An asymmetrical 3D/2D face recognition technique has been introduced. It is based on geometric invariants [5, 6], studied for the pose invariant object recognition problems for a long time. The crucial problem of choosing the control points, in case of faces, for the 2D image and 3D model based invariants has also been addressed.

The experiments have finally been made in order to assess the robustness of the method respect to the changes in pose and to the accuracy of locating the controls points. The results are encouraging in terms of recognition rate. Further works, can study the use of some other invariants, comparing their discriminating power and also integrating the texture information of faces within a multimodal framework.

References

1. Volker Blanz and Thomas Vetter, Face Recognition Based on Fitting a 3D Morphable Model. In IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, pp. 1191–1202, 2002.
2. M. Bronstein, Michael M. Bronstein, and Ron Kimmel, Expression Invariant 3D Face Recognition. In Proc. of Audio & Video-based Biometric Person Authentication (AVBPA), Lecture Notes in Computer Science 2688, Springer, pp. 62–69, 2003
3. Jennifer Huang, Volker Blanz, and Bernd Heisele, Face Recognition with Support Vector Macines and 3D Head Models. In First International Workshop on Pattern Recognition with Support Vector Machines (SVM 2002), pp. 334–341, 2002.
4. Fabio Lavagetto, Roberto Pockaj, The Facial Animation Engine: Toward a High-Level Interface for the Design of MPEG-4 Compliant Animated Faces, In IEEE Trans. on Circuits and Systems for Video Technology, vol. 2, no.2, march 1999.
5. Daphna Weinshall, Model-based invariants for 3D Vision. In International Journal of Computer Vision, vol. 10, no. 1, pp. 27–42, 1993.
6. Y.Zhu, L. D. Seneviratne and S. W. E.Earles, A New Structure of Invariant for 3D Point Sets from A single View., In IEEE International Conference on Robotics and Automation, pp. 1726–1731, May 1995.
7. Geometrix, Introducing FaceVision-The New Shape of Human Identification, http://www.geometrix.com/, 13 February 2005

3-D Face Modeling from Two Views and Grid Light

Lei Shi, Xin Yang, and Hailang Pan

Institute of Image Processing and Pattern Recognition,
Shanghai Jiaotong University, Shanghai, 200030, China
{sl0030322014, yangxin, panhailang}@sjtu.edu.cn

Abstract. In this paper, an algorithm for extracting three-dimension shape of human face from two 2D images using grid light is presented. The grid pattern is illuminated by incandescence light instead of laser in order to protect human eyes or skin and reduce cost. An uncoded grid pattern is projected on human face to solve the problem of correspondence between a pair of stereo images. Two images acquired at same time are smoothed to diminish noise at first. Then grid stripes from these images are extracted and thinned by a marked watershed algorithm. A new method based on graph connectivity to locate and label grid intersections from these images is also presented. According to labeling principles, a set of matched points is build. The set of matched points are further used to calculate three-dimension-depth information of human face. Experiment results show the feasibility of the proposed method.

1 Introduction

Face modeling is an important problem in many multimedia applications, such as teleconferencing, virtual reality, animation and face recognition. There are several major approaches for face modeling.

For instance, DeCarlo et al.[1] used the anthropometric methods to build a facial model; At first, a manually-constructed B-spline surfaces is constructed, and then surface fitting and constraint optimization is applied to the surface.

In [2], facial model is directly acquired from 3D laser scanners or structured light range sensors. Water's face model is a well-known model by this kind of equipments. In many face modeling methods, the facial model is regarded as a generic model. Kawai et al.[3] presented a method of range data integration based on region segmentation and extraction of feature parameters.

Also facial model is reconstructed by digital equipment, such as low-cost and passive input devices (video cameras or digital camera). For instance, Chen and Medioni [4] built facial model from a pair of stereo images. However, currently it is still difficult to extract sufficient information about the facial geometry only from 2D images. This is the reason why Guenter et al. [5] to utilize a large number of fiducial points to acquire 3D facial geometry.

Though we can acquire 3D information from expensive 3D laser scanners or structured light range sensors, it still takes too much time to scan and detected person must remain stable during the scanning. In order to overcome these problems, recently, some researchers try to incorporate some prior knowledge of facial geometry

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 994 – 1001, 2005.

or to make use of a generic facial model. For instance, Ansari, A.-N. et al. [6] deformed a 3D generic model from two orthogonal views (frontal and profile views) to acquire facial model. Zhang [7] deformed a generic mesh model to an individual's face.

In case a generic facial model can't be provided, some methods integrating structure light and computer vision is applied to human face modeling, such as Andrew Naftel et al. [8] and Philippe Lavoie et al. [9].

Philippe Lavoie et al. [9] proposed a new method for reconstructing 3D facial model from the left and right two-dimensional (2-D) images of an object using a grid of pseudorandom encoded structured light. The proposed method provided three distinctive advantages over a conventional stereo system:1) without new textures;2) less computational intensive;3) solving correspondence problem easily. We draw many inspirations from Philippe Lavoie's method.

Based on an uncoded grid light, our method allows for the introduction of a new procedure for the grid extraction and grid intersection location. The procedure can determine a set of points on the object surfaces on the left and right images with satisfied precision.

In the following, we will describe the main algorithms of this procedure. Section 2 deals with the design and extraction of the projected grid pattern. Section 3 is a description of the extraction process of grid information. 3D reconstruction of corresponding points from the right and left image is described in Section 4, and experimental results from different angles show the feasibility of our method. Eventually, in Section 5, a conclusion section reviews the main steps and the unique features of this system.

2 Uncoded Grid Light

In our method, using a projected grid pattern can capture the whole view of the face, instead of a line or a dot pattern in laser systems that require scanning to ensure covering the whole pattern. Then, a simple scheme is developed which fixes grid stripes along a pair of cross axes. The cross axes is easily distinguished from other stripes in the grid pattern. The uncoded grid light is shown in Fig. 1.

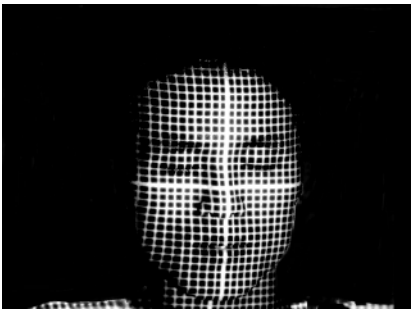


Fig. 1. Uncoded grid light

The process of extracting interest points is implemented in two steps: extracting the grid and extracting the intersections.

The extraction of grid stripe from the original image is also fulfilled in two steps: smoothing and extracting. The smoothing process consists of applying 2D Gaussian filter with standard deviation of $\sigma = 6.5$ (and kernel size 3×3). The filtering is used to eliminate noise signal in original image.

2.1 Watershed Algorithm

For most classical 3-Dmodeling methods [9-12,24-28], the projected grid stripes is illuminated by laser because laser is by far the most perfect light source for 3D modeling. The grid stripes can be easily extracted according to brightness, texture or color. The extracted stripes will be thinned by shrinking each stripe to its skeleton in[10,11] or thinning on binary images[12,13,27,28]. In [14], the process is also performed by a least square method approximation of the center points of the lines. Moreover, in [24-26], extracting line edge and line fitting is used to deal with the problem.

But the quality of our grid stripes is worse than classical structure light system because the grid light is illuminated by incandescence instead of laser in our system. The classical methods of extracting and thinning can't meet our requirements. In [9], Philippe Lavoie et al. applied directly the watershed algorithm to extracting and thinning grid stripe because grid stripe has illuminated by laser in this case.

Considering these conditions, our grid extraction process starts with the application of a watershed algorithm [15-17]. The watershed algorithm is applied on the gradient magnitude image. A fast immersion based on the algorithm developed by Vincent and Soille[15] is employed. All pixels in the gradient image are sorted by increasing gray-level values. Once the image is completely flooded, the watershed lines will be obtained. The lines from the watershed algorithm have exactly one pixel wide. Thus, the thinning process is also performed at the same time.

The watershed algorithm can find contiguous edges in an image accurately but suffers from the over-segmentation problem. Consequently, we may obtain many extra details that don't need when grid stripe is segmented, as shown in Fig 2(a). The original image is shown in Fig. 1. There are many methods to deal with over-segmentation, such as prior information [18], classifier [19] and marker [20-22] and so on. But in this paper, selected markers are employed to overcome the problem.

2.2 Marker Watershed Algorithm

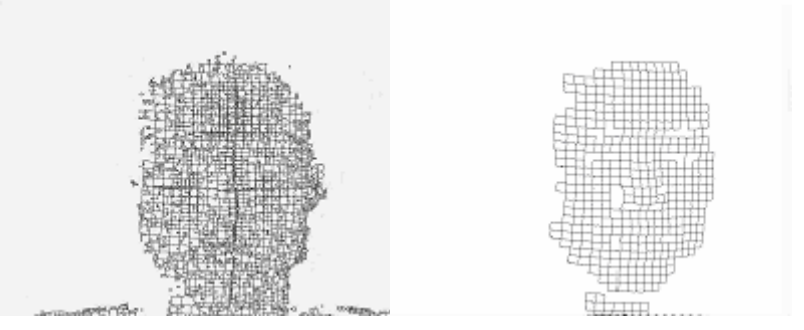
The marker-based watershed algorithm is a very efficient means for image segmentation and has been used in recent years. In this algorithm flooding starts directly from the marker instead of minima basins. There are some methods to choose right marker. For instance, in [20], S.Beucher et al. applied the marker-controlled watershed to road segmentation and obstacle recognition. These markers have been introduced by hand. Meyer's watershed algorithm [21] floods from two markers so that the final watershed line is located at the highest-crest line only. The result is a more meaningful or a more visually sensible segmentation. Hai Gao et al.[22] extracted marker based on luminance and color information.

Considering these methods and aiming at our images, a marked watershed algorithm is applied in this paper. As shown in Fig.1, the gray intensity of grid stripes is larger than other parts of original images. In order to remove those unuseful details, we flood directly from edge of grid stripe instead of the original image. The marker image can be expressed as followed:

$$\begin{cases} I(x,y) \in M, if\,(I(x,y) < Threshold) \\ I(x,y) \notin M, \text{otherwise} \end{cases} \tag{5}$$

Where *I(x,y)* is gray-level intensity of the original image, and *x*, *y* is pixel coordinate; *M* is the set of markers; *Threshold* is gray-level threshold of pixels in original images. In order to save time and remove unuseful details, the threshold should be selected to approximate the minimum intensity of grid stripes in original image. It can be acquired by Ostu's method [23].

Beginning from the marker image flooding is performed. The right watershed lines will be obtained. The original image is shown in Fig. 1. The extracted grid is shown in Fig. 2 (b) when selected markers are applied.



(a) (b)

Fig. 2. Extracted grid

3 Extracting Grid Information

More grid information, such as the precise location of these intersection points, their connectivity and their label, are needed after the grid has been exacted by the marker watershed algorithm. To obtain all information, some steps are required.

3.1 Intersection Detection and Location

For most classical methods of grid location [9-14,24-28], in general, suitable projector's angle is selected in order to deal with the problem when we have to extract intersections, such as [25,26]. In some cases, a conditional set that includes all cases on the extracted grid is proposed. Only those intersections that satisfied one case can

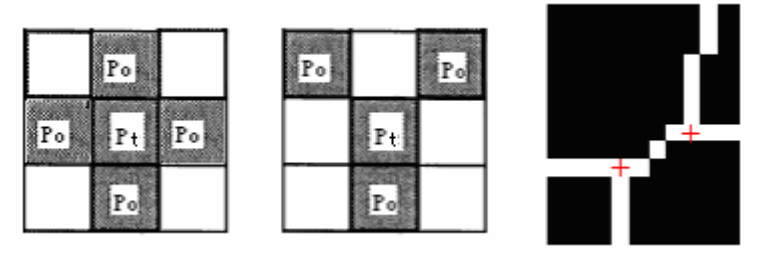
be regarded as candidates. In those cases, a template whose size is 3 by 3[9,12-14] or 7 by 7[10] is applied. Pixels with special location can be regarded as candidates in the template.

We have presented a new algorithm based on graph connectivity. Define candidate *Pt* be those pixels lying on the extracted grid stripes (Fig.3). Based on a set of conditions on the set of nonzero pixels in a 3 by 3 square neighborhood centered about the candidate points, these intersections are detected. We call these nonzero pixels the border point *Po*, as shown in Fig.3.

Firstly, our algorithm requires that the set X of candidates consists of three or four connected border points *Po* in the square neighborhood, shown in Fig.3(a) and (b).

If two intersection points *Pt* are connected, they are treated as one class. Then the average coordinates $\overline{Pt}$ of these intersections *Pti*(*i*=1…n, where n is the number of intersections from the same class) and the Euclidean distances between the average point $\overline{Pt}$ and all intersections *Pti* (*i*=1…n) of a class are calculated. The intersection *Ptj* with minimum Euclidean distance in the class *Pti* (*i*=1…n) will be regarded as the candidate *Pt*.

For flat areas, there is one intersection for each node of grid stripes. But for curved areas, there may be more than one intersection from the same node (shown in Fig.3(c)). Secondly, we will calculate *P3((x1+x2)/2,(y1+y2)/2))* if two intersections *P1(x1,y1), P2(x2,y2)*from the same node are so close to each other that the distance between them is no more than 5 pixels, then will replace these intersections by new candidate *P3*.



(a) (b) (c)

Fig. 3

3.2 Intersection Labeling

A key step in the 3-D reconstruction using grid light is to label these intersections in each 2D image (the stereo correspondence problem). For classical methods [9-14,24-28], coded grid pattern is used to deal with 3-D reconstruction problem according to pattern [9-12,24,28] or color [13,14,24-27]. This leads higher cost. We solve the problem by labeling all grid intersections before matching the left and right images. In [9], a labeling methods based on line can be found out. Labeling methods based on point is applied in [10]. We present a labeling method integrating line and point labeling by three steps:

1) Locating the cross axes;
2) Finding out all grid intersections on the cross axes, and then labeling these grid intersections;
3) Labeling all grid intersections based on labeled grid intersections.

The first step is to locate the cross axes in the original image. We can locate very easily the cross axes because its gray intensity and stripe width is more than that of other stripes. The erosion and dilation are employed in our method.

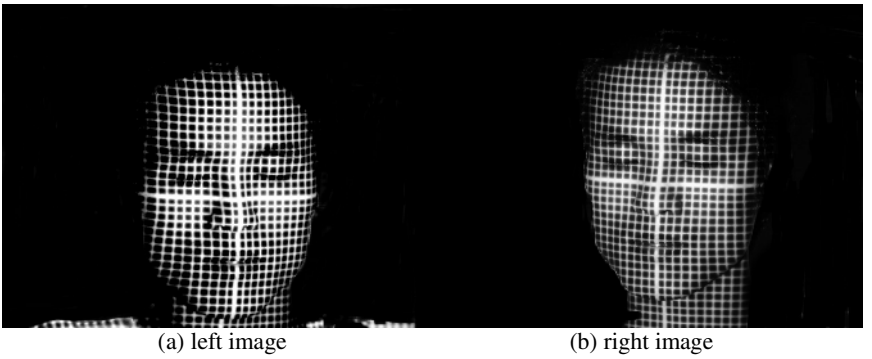
The second step is to find out intersections *Pt* on the cross axes. By comparing the sets *X* of all grid intersections with the locations of cross axes, we can find out that the set *Xc* of all grid intersections on the cross axes. The intersection of center *Pt* on the cross axes will be regarded as the principle point in the set *Xc*, and other intersections on the cross axes will be labeled referenced on the principle point *Pc*.

The final step is to label all grid intersections based on these labels of labeled grid set *Xc* on the cross axes. Since we have labeled the grid intersections on the cross axes, labeling set of grid intersections *X* can be obtained by using the method in [9].

4 Calculating 3D Information

In most classical structure light system, 3-D information is acquired as a result of a triangulation procedure. In order to avoid the risk of occlusion, binocular vision is applied to acquire 3D information of the object in our system.

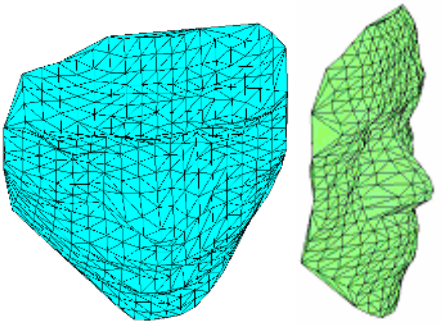
Original left image and right image are acquired from two digital cameras. The resolution of cameras is 1024×768 in our system. The two cameras' intrinsic parameters and the relative position between the two cameras can be acquired by calibration. The results from binocular vision will be used to show the feasibility of our system. Original right and left image are shown in Fig. 4.



(a) left image (b) right image

Fig. 4. Original images

3D face information is calculated by binocular vision. When two images of human face are taken, a 3D information map can be easily obtained. The disparity of a point gives a scaled version of its 3D information. The result from authors' system is shown in Fig.5.



(a) (b)

Fig. 5. The final 3D reconstruction results

5 Conclusion

In this paper, an algorithm to recover 3-D information of human face from two 2-D images of the same object is suggested by using incandescence light system consisting of a projector and two digital cameras. The grid pattern in our system facilitates the matching of the similar points situated on the two 2-D images. The resulted matched points help to determine the depth map.

References

1. D. DeCarlo, D. Metaxas, and M. Stone, An anthropometric face model using variational techniques. *In Proc. SICGRAPH,* pages 67-74, July 1998
2. F.I. Parke and K. Waters, Appendix **1:** Three-dimensional muscle model facial animation. *Computer Facial Animation,* September 1996
3. Y. Kawai, T. Ueshiba, T. Yoshimi, and M. Oshima, Reconstruction of 3D Objects by Integration of Multiple Range Data. *In Proc.11th International Conference Pattern Recognition*, I: 154-157, 1992
4. *Q.* Chen and G. Medioni, Building human face models from two images. *In Proc.IEEE 2nd Workshop Multimedia Signal Processing*, pages 117-122, December 1998.
5. B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, Making faces. In *Proc. SIGGRAPH*, pages 55-66, July 1998.
6. A-Nasser Ansari, 3D face modeling using two orthogonal views and a generic model. In *Proc. International Conference on Multimedia and Expo*, 3:289-92, July 6-9, 2003
7. Z. Zhang, Image-based modeling of objects and human faces. *In Proc. SPIE*, Volume 4309, January 2001
8. A. Naftel and Z. Mao, Acquiring Dense 3D Facial Models Using Structured-Light Assisted Stereo Correspondence, *Technique Report*, Department of Computation, UMIST, 2002
9. P. Lavoie, D. Ionescu, and E. M. Petriu, 3-D Object Model Recovery From 2-D Images Using Structured Light. *IEEE Transactions on Instrumentation and Measurement*, 53(2):437-443, April 2004
10. Stanley M. Dunn, Richard L. Keizer and Jongdaw Yu, Measuring the Area and Volume of the Human Body with Structured Light, *IEEE Transactions on Systems, Man And Cybernetics*, 19(6): 1350-1354, Nov/Dec 1989,

11. Petriu, E.M.; Sakr, Z.; Spoelder, H.J.W.; Moica, A., Object recognition using pseudo-random color encoded structured light, *In Proc. the 17th IEEE Instrumentation and Measurement Technology Conference*, 3:1237 - 1241, 1-4 May 2000
12. Guisser, L. Payrissat, R. Castan, S, A new 3-D surface measurement system using a structured light, *In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 784-786, 15-18 Jun 1992
13. G. Hu , G. Stockman, 3-D Surface Solution Using Structured Light and Constraint Propagation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(4):390-402, April 1989
14. P. Lavoie, D. Ionescu, and E. M. Petriu, A high precision 3D object reconstruction method using a color coded grid and nurbs, *In Proc. the International Conference on Image Analysis and Processing*, pages 370--375, Venice, Italy, September 1999
15. V. Luc and P. Soille, Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583-598, 1991
16. Paul R. Hill, C. Nishan Canagarajah and David R. Bull, Image Segmentation Using a Texture Gradient Based Watershed Transform, *IEEE Transactions on Image Proces*s, 12(12):1618-1634, 2003
17. Roerdink and Meijster, The Watershed Transform: Definitions, Algorithms and Parallelization Strategies, *FUNDINF: Fundamental Information*, 41:187-228, 2001
18. Susan Wegner, Klaus-Peter Pleissner, Helmut Oswald and Eckart Fleck, Hierarchical watershed transformation based on a-priori information for spot detection in 2D gel electrophoresis images, *In Proc. SPIE Int. Soc. Opt. Eng.* 3661:749,1999
19. Thierry Géraud, Pierre-Yves Strub, and Jérôme Darbon, Color Image Segmentation Based on Automatic Morphological Clustering, *In Proc. IEEE International Conference on Image Processing (ICIP'2001),* 3:70-73, Thessaloniki, Greece, October 2001
20. S.Beucher and M. Bilodeau, Road segmentation and obstacle recognition by a fast watershed transformation, *Intelligent Vehicles Symposium'94*, pages 296-301, 0ctober 1994
21. F. Meyer, Color Image Segmentation, *4th IEEE Conference on Image Processing and Applications*, 354:53, pages 303-306 (1992)
22. Hai Gao; Wan-Chi Siu; Chao-Huan Hou, Improved techniques for automatic image segmentation, *IEEE Transactions on Circuits and Systems for Video Technology*, 11(12):1273 – 1280, Dec. 2001
23. N. Otsu, A Threshold Selection Method from Gray-Level Histograms, *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1): 62-66, 1979
24. D. Bergmann, New approach for automatic surface reconstruction with coded light, *In Proc. Remote Sensing and Reconstruction for ThreeDimensional Objects and Scenes*, 2572: 2--9. SPIE, August 1995.
25. G. Stockman and G. Hu, Sensing 3-D surface patches using a projected grid, *Computer. Vision Pattern Recognition*, pages 602–607, 1986
26. N. Shrikhande and G. Stockman, Surface orientation from a projected grid, *IEEE Trans. Pattern Anal. Mach. Intel.* , 11(6):650–655, 1989
27. J. Tajima and M. Iwakawa, 3-D data acquisition by rainbow range finder, *In Proc. International Conference on Pattern Recognition*, pages 309–313, 1990
28. H. S . Yang, K. L. Boyer, and A. C. Kak, Range data extraction and interpretation by structured light, Purdue Univ, West Lafayette, *In Tech. Rep.*, pages 199-205, 1984.

Face and Facial Feature Localization

Paola Campadelli*, Raffaella Lanzarotti**, Giuseppe Lipori, and Eleonora Salvi

Dipartimento di Scienze dell'Informazione,
Università degli Studi di Milano,
Via Comelico, 39/41 - 20135 Milano, Italy
{campadelli, lanzarotti, lipori}@dsi.unimi.it
http://homes.dsi.unimi.it/~campadel/LAIV/

Abstract. In this paper we present a general technique for face and facial feature localization in 2D color images with arbitrary background. In a previous work we studied an eye localization module, while here we focus on mouth localization. Given in input an image that depicts a sole person, first we exploit the color information to limit the search area to candidate mouth regions, then we determine the exact mouth position by means of a SVM trained for the purpose. This component-based approach achieves the localization of both the faces and the corresponding facial features, being robust to partial occlusions, pose, scale and illumination variations. We report the results of the separate modules of the single feature classifiers and their combination on images of several public databases.

Keywords: Face and feature localization, skin color model, Support Vector Machine (SVM).

1 Introduction

Face localization[1] is a crucial first step for many applications such as face recognition, face expression analysis, and face tracking; moreover all these applications require the identification of the main facial features (eyes, mouth, nose) either to normalize the image or to further process them.

Both face and feature localization are challenging because of the face/feautures manifold owing to the high inter-personal variability (e.g. gender and race), the intra-personal changes (e.g. pose, expression, presence/absence of glasses, beard, mustaches), and the acquisition conditions (e.g. illumination and image resolution).

These tasks are often solved making some restrictions on the input images (uniform background, fixed scale and pose, etc.), and thus restricting the domain of applicability of the system [1], [7], [14]. Even more, sometimes the localization is accomplished in a manual or semi-manual way. For example Zhang and Martinez presented in [13] a

* Work partially supported by project "Acquisizione e compressione di Range Data e tecniche di modellazione 3D di volti da immagini", COFIN 2003.

** Work partially supported by the PASCAL Network of Excellence under EC grant no. 506778. This publication only reflects the author view.

[1] We speak about *face localization* when input images depict only one subject in the foreground and about *face detection* when no assumption is made regarding the number of faces in the images.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1002–1009, 2005.

face recognition system in which the face localization and normalization is manually done on the AR database [8]. To our knowledge the most significant face localization work has been presented by Smeraldi and Bigun [10]: they tested their application on the XM2VTS image collection [5], obtaining in 97.4% of cases the precise localization of the three main facial features (eyes and mouth), and the localization of at least two features in the 99.5% of cases. The main drawback of this method is that it is scale and pose dependent, limiting its real usability.

We designed a system for face and facial feature detection in color images consisting of two modules which try to exploit respectively the advantages of feature invariant approaches [11,12] and appearance-based methods [9,6]. The first module searches for skin regions within the image exploiting their peculiar colors and further information that helps to characterize faces. This step determines a *Skin-Map* which represents the restricted search area to be referred by the subsequent steps. In the second module, two different SVMs (trained to recognize eyes and mouths respectively) are applied only in correspondence to the skin regions with the objectives of both discriminating between faces and non faces and to localize the eyes and the mouth, if any. In particular, if at least one facial feature is localized within a skin region we validate it as a face. Of course, the detection of one feature can be enough to validate a skin region, but our final objective is to detect all three if they are all visible. In fact we observe that a good result would be to detect two features, since they would allow to determine the face scale and even to foresee the position of the lacking one; the third feature could be looked for in a subsequent step exploiting the knowledge given by the first two.

In [2] we presented the skin detection module, and the validation step based on the eye SVM only. At that stage we obtained high performance on images of very high quality, like those in the XM2VTS; on images with complex background, and which differ in illumination, scale, pose, and quality we obtained about 90.9% of face detection, with 125 false positives with respect to 783 eyes present.

In this paper we focus on the mouth localization (section 2), and we discuss (section 3) on how the conjunction of the eye and mouth SVM outputs can help in several directions such as to rise the detection rate, to achieve a lower acceptance of false positives and to deal with feature occlusions.

2 Mouth Localization with SVM

The localization step is mainly based upon the output of a statistical classifier, without taking into account any strong geometric knowledge of what constitutes a face. The only *a-priori* knowledge we exploit regards the peculiar color of mouths: to greatly reduce the search area to give in input to the classifier, we select those sub-regions within the *Skin-Maps* which show the peculiar mouth chromaticity (see section 2.1).

By searching for the mouth, we account for certain problematic situations that can happen in generic scenes, as occlusions of features other than the mouth or significant rotations of the head around the three axis. In fact rotations may greatly modify the face pattern while leaving substantially untouched the mouth appearance[2]. Moreover

[2] As specified in section 2.2, the SVM has some knowledge of examples extracted from portraits of people whose head is rotated to some extent along any degree of freedom.

our component based approach allows to treat some basic non-neutral expression, as anger, happiness and so on. We included in our treatment both mouth closed (neutral or angry) and slightly open (as if reading or smiling), as specified in the section 2.2, where we present the construction of the SVM classifier. In section 2.3 we give a brief description of the localization technique and finally in section 2.4 we summarize the results obtained on several data sets.

2.1 Mouth-Map

As we mentioned in the introduction, for each image we determined a *Skin-Map* (Fig. 2) which consists of one or more connected regions. Being focused on the mouth research, we reduce the search area furthermore on the basis of the peculiar mouth chromaticity. To this end we first apply a transformation to the pixels within the *Skin-Map* in order to simplify the segmentation task. Such transformation refers to the chrominance components in the *YPbPr* color space[3], and produces a *Mouth-Map* in which the mouth pixels assume high values:

$$Mouth\text{-}Map = Pr^2 \cdot (255 - (Pr - Pb^2))$$

The first factor of the formula exploits the consideration that usually mouth pixels have high values in the *Pr* plane; the second factor penalizes skin pixels which also have high *Pr* values but low *Pb* ones in comparison to the mouth pixels.

The *Mouth-Map* is thresholded considering each *Skin* region separately. Experimentally we observed that the mouth area corresponds roughly to the 3% of the face area and thus we selected in each skin region the 3% highest *Mouth-Map* values (Fig. 3).

Fig. 1. the input image

Fig. 2. The *Skin-Map*

Fig. 3. The *Mouth-Map*

On the 2147 images of the test set (for details see section 2.4) the *Mouth-Map* usually contains at least one region in correspondence to the mouth; such regions are

[3] *YPbPr* color space is obtained from the RGB components by means of a linear transformation which allows to separate the luminance *Y* from the chrominance *Pb* and *Pr*. Comparing several color spaces we retain this one since it has shown to be the most adapt to highlight both the skin and the mouth regions.

always strictly included in the portion of the image depicting the mouth. When the mouths are open, the lower and upper lips are very often disconnected, and the lower one is generally better defined, being the lower lip thicker on most subjects.

2.2 Training the Classifier

The objective of gathering a rich and representative training sample is accomplished by considering different image databases, each contributing to the definition of a specific aspect of the mouth pattern:

1. 585 images from the **XM2VTS** [5]: model the general, well-defined mouth pattern;
2. 525 images from the **AR** [8]: contain portraits under poor illumination conditions; we chose four sessions for each subject, referring to four different situations: neutral, anger, and smiling expressions, and non uniform illumination;
3. 890 images from the **Color FERET** [4]: suitable to model mouth belonging to faces rotated from -45° up to 45° around the vertical axis;
4. 208 images from the **BANCA-Adverse** [3]: useful to model the class of mouths taken from people who are reading (hence bending down and slightly opening the mouth);
5. 480 images from the **DBLAIV**: this selection helps to include in our classifier some knowledge about real world pictures. For instance it allows to model mouths taken from tilted faces[4] and it enriches the class of negative examples due to the high complexity and variety of the backgrounds.

The mouth classifier is based exclusively on the intensity information of the patterns, therefore all these images have been converted to gray scale prior to example extraction. For each image we dispose of manually placed ground truth: the coordinates of the eyes' centers, the nose tip and the four corners of the mouth (see Fig. 4). The sample is built by extracting from each picture the mouth (labelled as positive example), four non-mouth components chosen randomly out of twelve (see Fig. 5) plus three random examples taken from the background (or generally speaking from the complement of the mouth bounding box). These latter seven examples are labelled as negatives. The dimensions of the window used for extraction are related to the mouth width. We centered the mouth pattern on the lower lip for two main reasons. Firstly, if we consider the exact mouth center we would experience a greater variability of the pattern appearance due to the high variability of mouth expressions. On the contrary our positive examples show good uniformity in the lower half of the pattern. The second reason comes from the properties of the *Mouth-Map* that, especially if the mouth is not tightly closed, tends to be more accurate on the lower lip. However we did not include in our sample all mouth examples, since we wish to exclude from our model the subcase of open mouth[5]. By

[4] This subset of the DBLAIV contains faces whose *tilt* angle (defined as the angle between the vertical axis of the head and the horizontal plane) varies continuously from -60° to 60°, with mean around 0° and standard deviation 15°.

[5] In a preliminary experiment we trained a SVM on all mouth examples, but we obtained very poor results, meaning that the patter was too rich.

considering the distribution of the ratio width/height of each mouth, we observe that the vast majority of the examples we wish to treat falls above the value 2. This means that in general a mouth closed or slightly open is at least twice wider than tall. Hence, following our intentions, we characterized the positive class by discarding mouths whose ratio is under 2. This elimination step leaved us with 2353 positive examples.

The whole sample was then split into training and test set with proportion two third and one third respectively. This procedure gave rise to sets of cardinality 13340 and 6677. After the extraction, all sub-images have been contrast stretched and pyramidally down-sampled to the size of 16×16 pixels.

As in our previous work on eyes [2], we studied the positive class in terms of wavelet coefficients in order to reduce the number of features to consider, while retaining the essential pattern information, thus simplifying the training task. The feature selection process saved 95 wavelet coefficients for each example.

We trained a 1-norm soft margin SVM with Gaussian kernel and $\gamma = \frac{1}{2\sigma^2} = 4 \times 10^{-4}$, $C = 9$. Such parameters have been chosen as trade off between error reduction and generalization ability. By doing so, we selected a machine based on 1476 support vectors and achieving the 2.3% error on the test set. These results show a very accurate learning of the pattern by the SVM and make it suitable for the robustness of the following step.

Fig. 4. The arrows indicate the ground truth

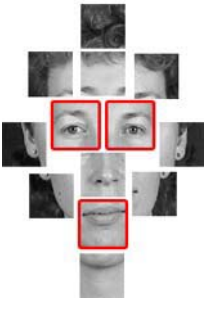


Fig. 5. Facial features

2.3 Localization Technique

The localization technique searches for mouths within the *Skin-Map*, with the idea of classifying as 'face' the region that corresponds to the detected mouth.

The mouth research poses two major problems. First, it is necessary to reduce the number of points to consider for classification, while not excluding the ones corresponding to the mouth. Second, the absence of any assumption on the scale of the face forces the mouth research on a range of possible dimensions. The two questions have implications both on the computational cost and on the accuracy of the technique.

Regarding the first issue, we have partially answered to it introducing the *Mouth-Map*, which reduces the research to the 3% of the *Skin-Map* area. A further reduction consists in selecting a proper set of candidates; to this end, we consider each region within the *Mouth-Map* separately: we determine its bounding box, and we enlarge it

in the lower part of the 50% of its height[6]. Afterwards, on the basis of the bounding box width w, we determine the scan step: the candidates are the points included in the enlarged bounding box that correspond to the interceptions of a grid of lines, spaced horizontally and vertically according to the scan steps $s_h = w/(5+k)$ and $s_v = w/(3+k)$ respectively[7], being k a constant that regulates such detail (in our experiments $k = 4$).

In order to evaluate each candidate point, we need to extract an example at a scale which fits the mouth model used to train the SVM. To this end, we infer the size of the examples on the basis of both the *Skin-* and *Mouth-Map*; these two contributes are quite independent and allow to determine a reliable approximation. To make it even more robust, we account for possible errors of over or underestimation of them, which means to consider different possible dimensions for mouths (hypothetically) present in the region. For simplicity, besides the optimal size d, we extract only two additional examples of sizes $(0.7 \times d)$ and $(1.3 \times d)$.

Let us call $\mathbf{x}_p$, $\mathbf{x}_p^-$ and $\mathbf{x}_p^+$ the examples corresponding to the same candidate p at the three different scales; we evaluate the *strength* of p by summing the margins of all three examples. This interpretation is not standard; usually the only output considered is the class label the SVM attributes to the example fed in, which corresponds to the sign of its margin. However, since the margin is proportional to the Euclidean distance of the example from the decision hyperplane, we treat it as a "measure" of the confidence with which the SVM classifies the example. Thus we define the function

$$f(p) = SVM(\mathbf{x}_p) + SVM(\mathbf{x}_p^-) + SVM(\mathbf{x}_p^+)$$

where $SVM(\mathbf{x}) = 0$ defines the optimal separating hyperplane. Being the three scales quite close, we usually observe a good correlation among the margins on positive examples, and the definition of f is useful to prevent the exclusion of a candidate due to a wrong *Skin-* or *Mouth-Map* estimate and simultaneously to weaken the strength of a pattern that looks similar to a mouth only at a certain scale.

Finally, given all the $f(p)$, we localize the mouth in two steps: at first we compute the score of each region adding all the $f(p)$ corresponding to it; the region with the highest score is identified as the mouth; afterwards, the mouth position is determined computing the centroid of all the points corresponding to the validated mouth region whose $f(p)$ is positive.

2.4 Experimental Results

We list here the results of face and feature localization; the experiments have been carried out on the same databases that we used to build the training sample but on disjoint image sets. Table 1 shows both the localization rate of eyes and mouth separately, and their combination which gives a better estimate of the overall behavior.

We observe that in the 10% of the data set one or more features are not available for classification, due to one of the following reasons: the incompleteness of the *Skin-Map* (if it does not cover all the features present); mouth occlusion by moustache or beard;

[6] This choice is driven by the fact that we trained the classifier to recognize the lower lip.

[7] We adopted two different scan steps for the horizontal and vertical position selections since we observed a higher sensitivity of the SVM to horizontal translations than to vertical ones.

eye occlusion because of strong rotations. This lack prevents to reach full localization of the three features on all images (the results in the second last column reflect this fact).

Since the loss of either one eye or the mouth can be easily recovered on the basis of the other two, in the last column we show the results obtained by relaxing the goal to the localization of at least two features.

Table 1. Face localization results

Localization results		**Eyes**		**Mouth**		**Face**	
Database	number of images	positive rate	false positives	positive rate	false positives	all three features	at most one feature missing
XM2VTS	583	97.9%	20	91.1%	15	87.8%	**99.0**%
AR	479	92.5%	75	86.6%	41	71.4%	**91.0**%
FERET Color	689	93.6%	163	88.3%	24	60.5%	**90.6**%
DBLAIV	189	80.2%	67	84.0%	19	62.4%	**84.1**%
BANCA Adverse	207	85.2%	58	86.3%	18	68.1%	**82.6**%

On the basis of the obtained results, we can conclude that the task of locating mouths seems more difficult than the one of detecting eyes; we think this is due to the fact that the eye pattern is more structured than the one associated to the lower lip, making easier and more robust the corresponding classification. Finally, we observe that if we consider the localization of just one feature as sufficient to validate the presence and position of the face, than we reach almost 100% of face localization on all databases; we notice that the addition of the mouth detector has increased of about the 10% the overall performance, considering that, using the eye detector only, we reached the 90% of face localization [2].

3 Conclusions

In this paper we presented a module for mouth localization based on a SVM classifier trained to recognize closed or slightly open mouths. This module is a part of a more general face and facial feature localization system which first localizes skin regions (on the basis of the peculiar skin color), and then searches for eyes and mouths. Such system is robust to partial occlusions and to changes in pose, expression and scale.

The results obtained prove that our system performs well: on the high quality images of the XM2VTS we obtained performances comparable to the ones presented in [10], while being more general and pose and scale-independent; on images which differ in illumination, scale, pose, quality and background we obtained good performance proving the generality and robustness of the system.

Differently from many scale-independent methods [6], which scan the image several times, we limit our search only to three different scales and on a small subset of points, exploiting the information given by color. Regarding execution times, our Java method,

run on a Pentium 4 with clock 3.2 GHz, takes approximately 5*ms* for each candidate point, bringing to a mean time of roughly 6 seconds to localize the facial features in images of 800×600 pixels.

In order to achieve a lower acceptance of false positives and a more precise feature localization, we intend to add a further module which takes into account the outputs of the different SVMs. Moreover we are working on the training of a classifier which will recognize open mouths; this will allow to apply the system to a even larger domain of images.

References

1. J.D. Brand and J.S.D. Mason. A skin probability map and its use in face detection. *Proceedings of International Conference on Image Processing*, 2001.
2. P. Campadelli, R. Lanzarotti, and G. Lipori. Face localization in color images with complex background. *Proceedings of the IEEE International Workshop on Computer Architecture for Machine Perception (CAMP 2005), Palermo, Italy. To appear*, 2005.
3. The BANCA database. Web address: http://www.ee.surrey.ac.uk/Research/VSSP/banca/.
4. The FERET Database. Web address: http://www.itl.nist.gov/iad/humanid/feret/. 2001.
5. The XM2VTS Database. Web address:http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/. 2001.
6. B. Heisele, P. Ho, J. Wu, and T. Poggio. Face recognition: component-based vesus global approaches. *Computer Vision and Image Understanding*, 91:6–21, 2003.
7. O. Jesorsky, KJ Kirchberg, and RW Frischholz. Robust face detection using hausdorff distance. *Lecture Notes in Computer Science*, 2091:212 – 227, 2001.
8. A.M. Martinez and R. Benavente. The ar face database. CVC 24, June 1998.
9. E. Osuna, R. Freund, and F. Girosi. Training support vector machines:an application to face detection. *Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR'97*, 1997.
10. F. Smeraldi and J. Bigun. Retinal vision applied to facial features detection and face authentication. *Pattern recognition letters*, 23:463–475, 2002.
11. J.C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance for the automatic detection of human faces in color images. *Proceedings of the IEEE International conference of Face and Gesture Recognition*, pages 54–61, 2000.
12. M. Yang and Narendra Ahuja. Gaussian mixture model for human skin color and its applications in image and video databases. *SPIE Proceedings Storage and Retrieval for Image and Video Databases VII, 01/23 - 01/29/1999, San Jose, CA, USA*, pages 458–466, 1999.
13. Y. Zhang and A.M. Martinez. Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class. *Proceedings of International Conference on Pattern Recognition (ICPR), 2004*, 2004.
14. J. Zhou, X. Lu, D. Zhang, and C.Wu. Orientation analysis for rotated human face detection. *Image and Vision Computing*, 20:239–246, 2002.

Multi-stage Combination of Geometric and Colorimetric Detectors for Eyes Localization

Maurice Milgram, Rachid Belaroussi, and Lionel Prevost

LISIF Université Pierre et Marie Curie BC 252
4, Place Jussieu, 75252 Paris Cedex 05, France
maurice.milgram@upmc.fr,
rachid.belaroussi@gmail.com,lionel.prevost@upmc.fr

Abstract. We present in this paper a method for the localization of the eyes in a facial image. This method works on color images, applying the so called Chinese Transformation (CT) on edge pixels to detect local symmetry. The CT is combined with a skin color model based on a modified Gaussian Mixture Model (GMM). The CT and the modified GMM give us a small rectangular area containing one eye with a very high probability. This rectangle is then processed to find the precise position of the eye, using four sources of information: a darkness measure, a circle finder, a "not skin" finder and a position information. Experimental results on a large database are presented on nearly 1000 faces from the ECU database.

1 Introduction

We present in this paper an algorithm for locating the eye pair in a facial image. Face detection/localization has become a very important issue for applications such as videoconferencing, biometrics, human detection, digital picture retrieval, etc. This issue is connected to several sub-problems: detection of facial landmarks (eyes, mouth, etc..), gaze detection and tracking, pose estimation, facial expressions analysis.

Detecting faces often starts with the detection of face appearance (pattern) or color, normalizing the image to take into account illumination and geometrical variations. Facial landmarks bring useful information to cope with the huge variability of face appearance. In this spirit, eye detection is a very important step in face detection. It can help to confirm or reject a hypothesis, decreasing the false detection rate.

We find two generic approaches in eye detection:

- Binary decision where the question is: "is this precise scanning window an eye or not?". In this approach, the problem of detection is broken into a large set of decision problems.
- Localization where ones try to locate eyes in a face image (or sub-image), supposing that this image contains at least one visible eye. In this more holistic approach, we can use global information.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1010 – 1017, 2005.

Our method belongs to the second family but can also make use of some local properties.

Huang and Wechsler [8] used wavelets to represent eyes and RBF networks to perform the classification of facial regions as eye / non-eye. Gabor wavelets are often used to process face images and iris coding [5]. In [3] on the detection of "genuine smiles", authors indicate that Gabor filters representations of images are highly effective for face recognition and expression analysis. Some authors [7] propose to extract feature points from faces with help of edge and corner points. They use classical image analysis techniques (binarization, projection, gray level morphology) to find these feature points in a pure bottom up way. A method to locate eyes in human face images using fractal dimensions is presented in [10]. Sometimes, authors try to find a specific feature inside the eye: [2] introduce a method to detect the white visible portion of the eyeball, the sclera which color is different from the surrounding area.

The paper is organized as follows. We present the database and the pre-processing in section 2. In section 3, we introduce our Chinese Transform. Next section is devoted to a probabilistic method for improving our detection and allowing an extrapolation of the second eye if only one has been detected. Section 5 describes a refinement technique for the y coordinate. Experimental results are presented in section 6. The last section is devoted to our concluding discussion.

2 Database, Pre-processing and Normalization

ECU database [11] contains about 3000 pictures with one face or more. Pictures are of various complexity, resolution and illumination. We used the first half of the database (training set) for our statistic estimations and parameters tuning. The whole database was used to get experimental results and no significant bias was observed between the two halves of the database. Face image, after the location of the head boundary, is transform in a grayscale image by simply averaging the three RGB values without taking into account of prior knowledge on the skin color like in [9]. This makes the method robust with respect to illumination variations and allows the treatment of monochrome pictures (for example close infra-red). We assume in our paper that the locating of the head boundary is done to focus on eye location. In order to normalize the images from the radiometric point of view, we chose a linear stretching of the gray-levels dynamic. This transformation gave better results than the well-known histogram equalization.

To find the orientation of edge pixel gradient, a circular mask is used for low pass filtering. Then, we classically determine the 2 components (Gx and Gy) of the gradient by using a simple Roberts operator. We are interested only in the direction of the gradient ($\theta = arctan(Gy/Gx)$). The gradient magnitude is only used to select edge pixels. It should be recalled that the gray level dynamic is always between 0 and 255 because of the initial stretching. The orientation θ is quantified on N values ($N=8$ for us). We will call $ORIENX$ the map of the quantified orientations thus obtained; this map can be divided into $N+1$ sets corresponding to orientations $0,1,...,N-1$ and one more set corresponding to non-edge pixels.

3 First Stage: Coarse Eye Localization

3.1 Chinese Transform

CT is a global transformation which input is the gradient orientations of the edge pixels of the image. It provides as output two feature maps (or cumulative arrays). The main feature map (so called *VOTE*) gives information on the centers of the eyes. The basic idea is that the eyes occupy, including the eyebrows, a zone which is darker than its environment and has an elliptic shape. Because of its property of approximate central symmetry, the eye center is the middle of many segments joining 2 opposite points of the boundary and these pairs of points having opposite gradient orientations (Fig 1. shows an example). The CT is quite different but related to the work presented in [12] on generalized symmetry. Unlike the Hough Transform, the feature space of the CT is simply the image. Although CT looks like the GHT (Generalized Hough Transform which uses the gradient), it strongly differs from it because one CT can detect at the same time ellipses of any size and eccentricity. It is based on the central symmetry property of ellipses.

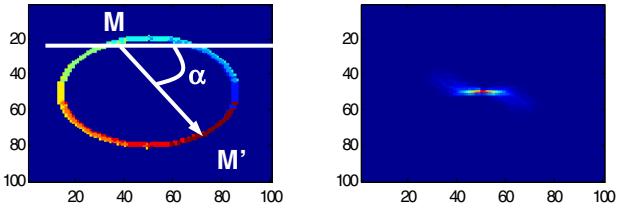


Fig. 1. Applying CT to an ellipse. Left: gradient orientation of edges. Right: resulting *VOTE*.

To start, the general principle of CT consists in determining, for each pair of pixels *(M,M')* with orientations *O=ORIENX(M),O'=ORIENX(M')* ranging within [*0 ; N-1*] if they are part of the voter set *VS* or not. In our case, the criterion of membership is based on the relative position of *M* and *M'* and on the condition *O =2* and *O' =6* (in other words *O*=North and *O'*=South). The relative position of the points *M(x, y)* and *M'(x', y')* must check the following geometrical constraints:

(1) $y-y' \geq \delta_{VERT}$

(2) $45° < \alpha < 135°$ where $\alpha = arctan\frac{y-y'}{x-x'}$

(3) $|| MM' || < \delta_{DIST}$

These 3 conditions indicate that (1) M is above M', (2) the vector MM' is not horizontal and (3) points M and M' are not too distant. One can notice that the first condition does not involve the second one since the points can be relatively distant on the horizontal axis. The values $\delta_{VERT} = 0.08H$ and $\delta_{DIST} = 0.25H$ are taken from experimental results (the parameter *H* is the image height). We define and initialize to zero two arrays *VOTE* and *DIAM* which have the size of *ORIENX* (orientation map). For each pair *(M,M')* of elements of the voter set *VS*, we increment the value of *VOTE(P)* where *P* is the middle of the segment *[M,M']*. The CT name comes from

the fact that China is sometimes called the "empire of the middle". To preserve an information about the distance between the 2 points voters for a point *P*, we store in *DIAM(P)* (for diameter) the average of the distance *d(M, M')* where *P* is the middle of *[M, M']* (the CT name comes from the fact that China is sometimes called the "middle empire"). This information will be also used to determine the size of a rectangular window around each eye. The result provided by the CT is thus a pair of maps: *VOTE* and *DIAM*. True eyes locations correspond fairly well to the pixels with maximum number of votes. To eliminate most of parasitic local maxima, we carry out a smoothing of the *VOTE* map. Nevertheless, we need additional cues to find accurately the eyes location.

3.2 Modified GMM Skin Color Model

The skin color is often used as a cue for detecting, localizing and tracking faces. Several classes of skin color models have been proposed: non parametric, parametric and neural. GMM belongs to the category of parametric estimators and are very popular, especially for the skin/no-skin discrimination. In [4,6], authors discuss the choice for the best chromatic space (RGB, HS, YcbCr, YIQ, CIE XYZ, ab, etc..), the training algorithm (most of them use the Expectation-Maximization algorithm) used to learn parameters (weights, means and covariance matrix) and to determine the complexity (number of components).

We decided to learn a skin color model for each face, hence in a non-supervised approach. The idea is that in general, eyes are in the upper part of the face. We learn our parametric model on the lower part of the face and then apply it on the whole face, looking for pixels that are not well accepted by the model. EM algorithm has proved to be efficient when using a large amount of data for training. In our case, we prefer a simpler method because we have to deal with a small number of skin-pixels. This number is small because we work on one single image and also because we subsample (1 out of 9) the image to eliminate (or to decrease the weight of) irrelevant regions (mouth, nostrils, beard). So we use a simple K-means algorithm with Euclidean distance to get the best center of each Gaussian. With each center, we have a specific subset of training examples that are closer to this center than to the other. We estimate Gaussian parameters (means and covariance matrix) for each subset.

Several experiments lead us to choose the RGB colorimetric space that gives better results than the YCbCr one. The number of Gaussian was set to 3 and we perform 5 iterations of K-means with an initialization based on a uniform spatial repartition of pixels (after sub-sampling and working only on the lower part of the face image). Applying this model to the whole face image produces the *Skin_map* containing the Mahalanobis distance of each pixel (in the RGB space) to the best center found with the K-means algorithm.

3.3 Combination

We have now 2 maps: the Smoothed *VOTE* accumulator of the Chinese Transform and the *Skin_map*. We search first all local minimum of *Skin_map*. Let *LM* be this set. We believe that eyes locations are probably close to one of these minima. We simply take the point (*AM*) with the absolute maximum of the smoothed vote accumulator over *LM* pixels. We have found experimentally that a rectangle centered on the point

AM and with a width and height of *0.25ImSize* gives a probability of 99%. *ImSize* is the mean of the width and the height of the image computed other the whole training set. To find the rectangle for the second eye, we set to zero all the votes of the first rectangle in the vote accumulator; then we search again the absolute maximum of this modified array.

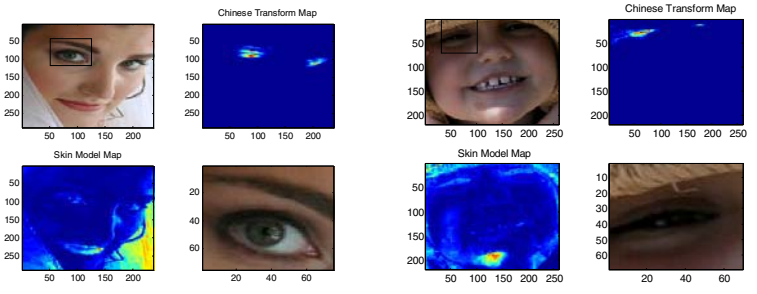


Fig. 2. Combination of CT with the modified GMM

4 Second Stage: Fine Eye Localization

We have found a small rectangle for each eye and we have now to refine this area to locate more precisely each eye. Computing 4 maps will do this:

4.1 Darkness Measure

Since we have already found a small area containing the eyes, we can exploit the fact that pupils are dark regions. Empirical tests showed that a threshold of 25% is good. We compute a darkness map with the formula (4), for each pixel *(i,j)* with gray level *g(i,j)*:

$$Dark\_map(i,j) = U(\delta - g(i,j)) + \frac{10U(g(i,j)-\delta)}{g(i,j)-\delta+10} \quad (4)$$

where $U(x)$ is the usual threshold function, δ is the value to get 25% of pixel gray levels below. *Dark_map* takes the value 1 for all pixels below δ; for other pixels, it takes a decreasing value as the gray level increases. With δ=55, *Dark_map*= 0.05 for white pixels (gray level=255) , and *Dark_map*= 0.5 for the gray level 65. So it is a fast decreasing function from the value 1 as the gray level increases from δ value.

4.2 Pupil Finder

We compute the classical Generalized Hough Transform for Circle (GHTC) using the intensity gradient orientation. Each edge pixel votes for the circle center in the opposite direction of the gradient, at a distance equal to the (hypothetical) radius. This GHTC is applied for each possible value of the radius. We find that the mean radius of the pupil is *0.14.L* where *L* is the width of the rectangle. So we take *R = 0.14L* and the set of possible radii is the interval *[0.7R 1.3R]*. After the voting process, values are normalized in the range [0 1] and give us the *Hough_map*.

4.3 Local Skin Color Model

The global skin model computed in the previous section is no more accurate for the rectangle. We compute again a local skin color model with sub-sampling (factor is set to ½) and with only one Gaussian. We obtain the *Skin_map*.

4.4 Position Information

Rectangle provided by the first stage of our method gives a rather centered position for the "true" position of the pupil. It means that there is more information provided by this step than a simple rectangle, implicitly representing a uniform density probability. We build the *Position_map* as 2D triangular function with value 1 in the center of the rectangle and 0 just on the boundary of the rectangle.

The 4 maps are simply multiplied to get a *Fusion_map*. It means that we consider these four maps equivalent and we make a kind of "*fuzzy logical and*" of these four information. Other kinds of fusions were tested but none outperformed the simple multiplication. The position of the center of the eye (see discussion below about the ground truth) is determinate as the location of the absolute maximum of the *Fusion_map* (figure 3) within each sub-window.

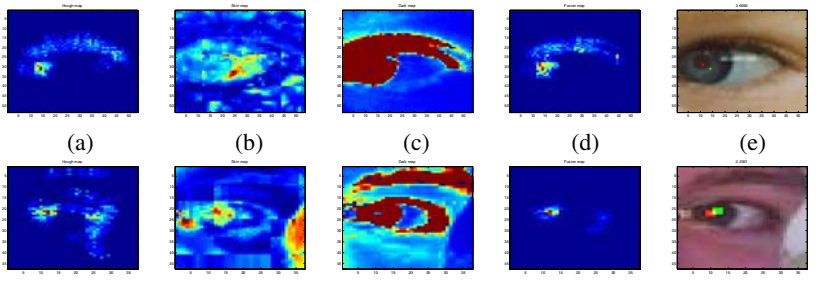


Fig. 3. Hough map (a), skin map (b), dark map (c), fusion map (d) and localization (e) (red: ground truth, green: localization)

5 Results and Discussion

Recall that we have taken all faces of the ECU database, even with closed eyes, glasses, hairs on face, unusual pose. We just select faces with more than 150 pixels width and height (32% of the faces are selected with a mean width of 218 pixels). The ground truth was done manually: operators click on the pupil's center of each eye, even if person are wearing glasses, even with reflections. We have checked this ground truth and it appears that there is a small dispersion among operator labelling. Moreover, our method is not really a "pupil finder" but rather a "eye finder". One can see that, sometimes, pupils are far from the eye center or even are not visible (closed eye). If the distance is smaller than 10 pixels (true with probability 0.9), it means that the ground truth is within a square of 14x14 pixels centered on the detection. This

value of 14 pixels is about 6.4 % of the average width of the selected faces. For these 923 faces, we compute the error i.e the distance (in pixels) between the ground truth and the detection provided by our algorithm. The histogram is presented (figure 4). We obtain several statistics for the error (table 1). We can summarize our results like this: for 90% of the faces, we accurately find the eyes and in 50%, we find exactly the pupil center. We present in figure 5 some examples of eye localization.

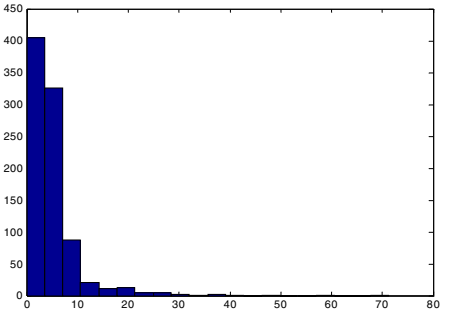


Fig. 4. Error (in pixel) histogram for 923 faces

Table 1. Localization error

Median	Mean	P(Error<10)	P(Error<15)	P(Error<20)
3.6056	5.0609	0.9147	0.9579	0.9761

We have done some experiments to evaluate the processing time. These measure have been done on 20 images with a MatLab 6.5 environment. The average computing time is 9.1 sec per image for the total detection, processor is a Pentium 4 at 2.6 Ghz.

6 Conclusion and Prospects

We have presented a system that detects and localizes precisely eyes in a facial color image. It is a two-stage combination of several detectors based on the colorimetric properties of face and eye and the geometric properties of eye and pupil. We introduce a new cumulative transform that can detect central symmetry. Experimental results show the accuracy of our method.

As we developed in our lab an automatic face localizer [1], eye detector should be integrated as a part of the whole system to help, confirm or reject a hypothesis in order to decrease the false detection rate.

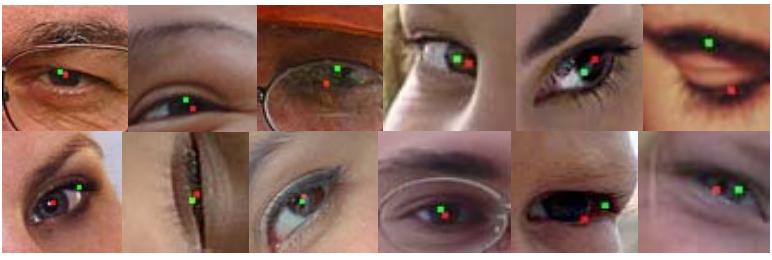


Fig. 5. Examples of localization (red: ground truth, green: localization)

References

[1] Belaroussi, R. , Prevost, L. & Milgram, M.: Combining model-based classifiers for face localization, to appear in Proc. of IAPR Conf. on Machine Vision & Application (2005)

[2] Betke, M., Mullally, W. & Magee, J.: Active detection of eye sclera, IEEE CVPR Workshop on Human Modeling, Analysis and Synthesis (2000)

[3] Braathen, B. and Bartlett, M. S. Littlewort-Ford G. & Movellan J. R.: 3-D head pose estimation from video by nonlinear stochastic particle filtering, UCSD MPLab TR 2001.05 (2001)

[4] Caetano, T.S. & Barone, D.A.C.: A probabilistic model for the human skin color, IEEE Int. Conf. on Image Analysis and Processing, Int. Conf. On Image Analysis & Processing (2001) 279-283

[5] Daugman, J.G.: Complete Discrete 2-D Gabor Transform by Neural Networks for Image analysis and Compression, IEEE Trans. ASSP, Vol. 36, n°7 (1988) 1169-1179

[6] Garcia, C. & Tziritas, T.: Face detection using quantized skin color regions merging and wavelet packet analysis, IEEE Trans. on Multimedia, Vol. 3, (1999) 264-277

[7] Gu, H., Su, G. and Du, C.: Feature points extraction from face, Image and Vision Computing New Zealand (2003)

[8] Huang, J. & Wechsler, H.: Eye location using genetic algorithm, Int. Conf. On Audio and Video-Based Person Authentification (1999)

[9] Hsu, R.L., Abdel-Mottaleb, M. & Jain, A.K.: Face detection in color images, IEEE Trans. PAMI, Vol. 24, n°5 (2002) 696-706

[10] Lin, K.H., Lam, K.M. & Siu, W.-C.: Locating the eye in human face images using fractal dimensions, IEEE Proc.Vis.Image Signal Process., Vol. 148, n°6 (2001)

[11] Phung, S.L., Bouzerdoum, A. & Chai, D., Skin segmentation using color pixel classification: Analysis and comparison, IEEE Trans. On PAMI, to be published in 2005

[12] Reisfeld, D. & Yeshuran, T., Robust detection of facial features by generalized symmetry, IEEE Int. Conf. on Pattern Recognition (1992) 117-102

Score Selection Techniques for Fingerprint Multi-modal Biometric Authentication

Giorgio Giacinto, Fabio Roli, and Roberto Tronci

Department of Electric and Electronic Engineering,
University of Cagliari, Piazza D'Armi, I-09123 Cagliari, Italy
{giacinto, roli, roberto.tronci}@diee.unica.it

Abstract. Fingerprints are one of the most used biometrics for automatic personal authentication. Unfortunately, it is often difficult to design fingerprint matchers exhibiting the performances required in real applications. To meet the application requirements, fusion techniques based on multiple matching algorithms, multiple fingerprints, and multiple impressions of the same fingerprint, have been investigated. However, no previous work has investigated selection strategies for biometrics. In this paper, a score selection strategy for fingerprint multi-modal authentication is proposed. For each authentication task, only one score is dynamically selected so that the genuine and the impostor users' scores distributions are mainly separated. Score selection is performed by first estimating the likelihood that the input pattern is an impostor or a genuine user. Then, the min score is selected in case of an impostor, while the max score is selected in case of a genuine user. Reported results show that the proposed selection strategy can provide better performances than those of commonly used fusion rules.

1 Introduction

Multimodal biometric systems have been proposed to increase the accuracy of authentication systems [4]. According to Prabhakar and Jain [10] multimodal biometric systems can be subdivided into five different scenarios: multiple biometrics [1] [5] [11], multiple acquisitions of the same biometry with the same sensor [6], multiple representations, and matching algorithms for the same biometry [8] [10], multiple units of the same biometry (for example different fingers) [6], multiple sensors for the same biometry [9]. In this paper we use fingerprints as biometrics.

The vast majority of multimodal systems are based on fusion strategies at the score level, so that the score of different matchers are combined to attain a "new" score. Typical fusion rules adopted for multimodal system are the min, max, median, mean, as well as trainable rules such as neural networks [9]. However, the output of different matchers could be exploited not only by fusion rules, but also by dynamic selection mechanisms. In the pattern recognition field, *selection* mechanisms have been proposed to select, for each input pattern, the classifier that provides the correct output [7]. This formulation of the selection problem

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1018–1025, 2005.

requires that each classifier outputs a class label for each input pattern. An example of selection mechanism is the algorithm for Dynamic Classifier Selection (DCS) that estimates the competence of each classifier by computing the local accuracy in a neighborhood of the test pattern [2] [12].

It is easy to see that the use of DCS mechanisms for multimodal biometric systems is not straightforward because: (a) it is hard to define a "local neighborhood" of the input pattern where accuracy should be computed; (b) the computation of the accuracy requires that for each matcher an acceptance threshold is set. It is worth pointing out that the vast majority of multimodal systems employs fusion rules at the score-level to avoid setting acceptance thresholds, as it is difficult to relate the choice of the thresholds of individual matchers to the performances of fusion rules at the decision level. On the other hand, fusion rules at the score-level aim at producing distributions of scores allowing for better discrimination between legitimate users and impostors.

In this paper we formulate a selection mechanisms for multimodal systems aimed at separating the distributions of scores of genuine and the impostor users. First, the available scores are used to predict the most likely state of nature of the input pattern, i.e. impostor or genuine. Then, the maximum score is chosen if the pattern is likely to be a genuine, while the minimum score is chosen if the pattern is likely to be an impostor.

The rest of the paper is organized as follows: Section 2 proposes an *ideal selector* for a multimodal systems. In Section 3 an algorithm based on the *ideal selector* is proposed. Experimental results on a fingerprint dataset are presented in Section 4. Section 5 concludes the paper.

2 Dynamic Score Selection for Multimodal Systems

In the field Multiple Classifier Systems, the *oracle* has been defined as the *ideal selector* which always selects, for each input pattern, the classifier that provides the correct label, if any. Accordingly, some algorithms have been proposed in the literature [3].

In a recent paper this definition of the *oracle* has been applied to multimodal systems where for each matcher an acceptance threshold has been set in advance [9]. By setting an acceptance threshold, each matcher can be considered as a classifier whose output is an accept/reject decision. As a consequence, an *oracle* has been defined as the ideal selector which selects the matcher, if any, that correctly authenticated or rejected the input fingerprint in the case of a genuine or impostor user, respectively.

In this paper we propose a different formulation of the selection problem, that does not require setting acceptance thresholds for the available matchers. Thus, each matcher is not considered as a classifier, and selection is performed at the score level, so that, for each input pattern, only one of the available scores is selected. Such a selection mechanism should produce a distribution of impostor and genuine scores separated as much as possible. To attain this result, such an *ideal selector* should choose, for each pattern, the largest score in case of patterns

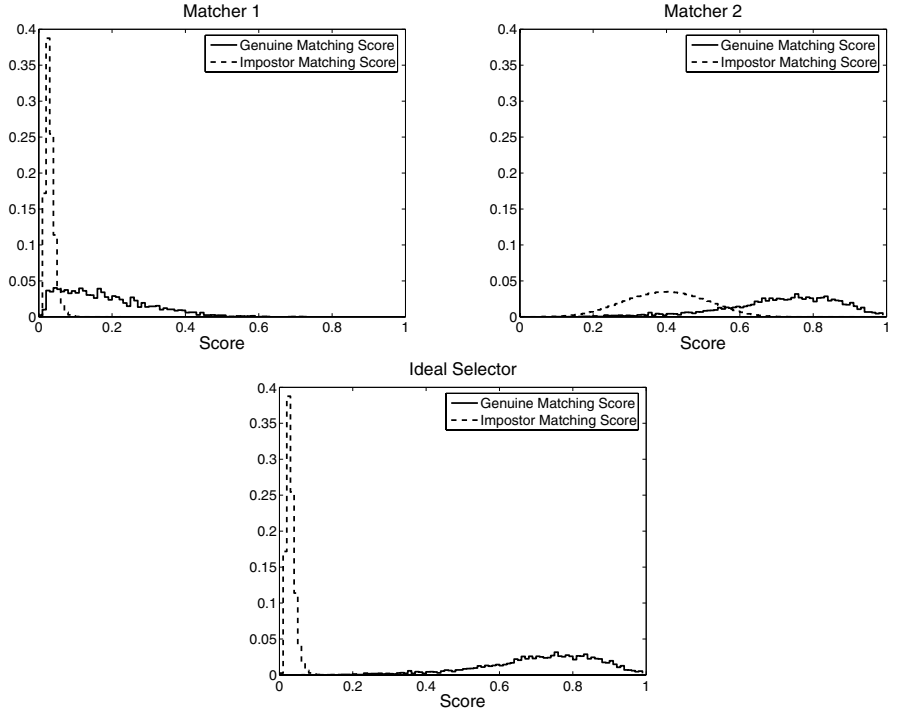


Fig. 1. An example of *ideal selector* with two matchers

belonging to genuine users, and the smallest score in case of patterns belonging to impostors. In other words, the *ideal selector* can be viewed as a switch between the *Max* and the *Min* fusion rules. Figure 1 shows an example of the selection rule implemented by the proposed *ideal selector*. It is easy to see that the use of the *Max* rule for genuine users produces a genuine score distribution shifted toward large score values, while the *Min* rule in case of impostors produces an impostor score distribution shifted toward small score values. In addition, with respect to the score distributions produced by the available sensors, the two distributions produced by the *ideal selector* exhibit a larger distance between the means, and a smaller variance.

In order to show why score selection should work, let us recall the two errors commonly used to evaluate authentication systems for a given acceptance threshold s^*, i.e. the False Acceptance Rate (FAR)

$$\begin{aligned} FAR(s^*) = P(s > s^* | \text{impostor}) = \int_{s^*}^{1} p(s|\text{impostor})\mathrm{d}s = \\ = 1 - \int_{0}^{s^*} p(s|\text{impostor})\mathrm{d}s \end{aligned} \tag{1}$$

and the False Rejection Rate (FRR)

$$FRR(s^*) = P(s < s^* | \text{genuine}) = \int_{0}^{s^*} p(s|\text{genuine})\mathrm{d}s \tag{2}$$

It can be shown that for a given value of FAR the *ideal selector* exhibits a FRR smaller than that of any of the available matcher, and that for a given value of FRR the *ideal selector* exhibits a FAR smaller than that of any of the available matcher. According to the definition of ideal selector, it can be easily seen that for a given value of the threshold s^*

$$\begin{aligned} FAR_{ideal\ selector}(s^*) &\leq FAR_j(s^*) \qquad j = 1, \ldots N \\ FRR_{ideal\ selector}(s^*) &\leq FRR_j(s^*) \qquad j = 1, \ldots N \end{aligned}$$

where N is the number of available matcher.

Thus it follows that, if the value of FAR is fixed, different thresholds have to be set, the threshold of the matchers being always larger than or equal to that of the *ideal selector*. As a consequence, the FRR of the matchers are always larger than the FRR of the ideal selector. The same reasoning can be used to show that for a fixed value of FRR, the FAR of the matchers is always larger than or equal to the FAR of the ideal selector.

3 A Score Selection Algorithm

Let us assume that a set of N matchers $M = \{M_1, M_2 \ldots M_N\}$ is available. Each matcher M_j outputs a score s_j for each pattern x. Moreover, let us use the Gaussian model for the genuine user and impostor distributions.

An algorithm for selection based on the *ideal selector* described in Section 2 should be made up of two steps:

1. Estimation of the more likely state of nature ω of the pattern x, where $\omega \in \{\text{genuine}, \text{impostor}\}$
2. Selection of the score s_{sel} based on the estimated state of nature ω as

$$s_{sel} = \begin{cases} \max_j(s_j) & \text{if} \quad \omega = \text{genuine} \\ \min_j(s_j) & \text{if} \quad \omega = \text{impostor} \end{cases}$$

In order to estimate the state of nature of pattern x, for each matcher M_j ($j = 1 \ldots N$), a measure of the expected error in classifying the pattern as being a genuine or an impostor is computed (Section 3.1). In the rest of the paper we will refer to this algorithm as Dynamic Score Selection (DSS).

3.1 Minimum Expected Error

Let us assume that for any value of s^*, the following relation holds

$$\int_0^{s^*} p(s|\text{impostor})\mathrm{d}s > \int_0^{s^*} p(s|\text{genuine})\mathrm{d}s \tag{3}$$

i.e., in the range $[0, s^*]$ impostors outnumber genuine users. This is usually true for a wide range of values of s^*.

In order to compute the expected error in assigning a pattern to one of the classes "impostor" or "genuine" given the output score s_j of the *j–th* matcher, let us set the acceptance threshold to s_j and compute the difference

$$D_j = |FRR(s_j) - FAR(s_j)| \tag{4}$$

By substituting Equations (1) and (2) in (4),

$$\left| \int_0^{s^*} p(s|\text{genuine})\text{d}s - 1 + \int_0^{s^*} p(s|\text{impostor})\text{d}s \right|$$

If $FRR(s_j) > FAR(s_j)$, then $\int_0^{s^*} p(s|\text{genuine})\text{d}s - 1 + \int_0^{s^*} p(s|\text{impostor})\text{d}s = D_j > 0$. According to the assumption in Equation (3) we have

$$\int_0^{s^*} p(s|\text{impostor})\text{d}s > \int_0^{s^*} p(s|\text{genuine})\text{d}s = D_j + 1 - \int_0^{s^*} p(s|\text{impostor})\text{d}s$$
$$\int_0^{s^*} p(s|\text{impostor})\text{d}s > \frac{D_j+1}{2}$$
$$FAR(s_j) = 1 - \int_0^{s^*} p(s|\text{impostor})\text{d}s < \frac{1-D_j}{2}$$

Thus it follows that by accepting the input pattern as a genuine user an error smaller than $(1-D_j)/2$ is expected. As a consequence the input pattern is likely to be a genuine user.

Analogously, if $FAR(s_j) > FRR(s_j)$, thus it follows that $FRR(s_j) < (1 - D_j)/2$, and the input pattern is likely to be an impostor. The highest the value of D_j computed according to Equation (4), the most likely the decision, as it leads to the minimum expected error. Summing up, the state of nature ω for pattern x can be estimated as follows:

1. let us compute, for each matcher M_j, the value of D_j
2. let $k = \text{argmax}_j(D_j)$
3. then

$$\omega = \begin{cases} \text{genuine} & \text{if} \quad FRR(s_j) > FAR(s_j) \\ \text{impostor} & \text{if} \quad FAR(s_j) > FRR(s_j) \end{cases} \tag{5}$$

The difference in Equation (4) can be estimated by assuming Gaussian distributions for the score of genuine users and impostors. Let μ_G (μ_I) and σ_G (σ_I) be the mean and the standard deviation of genuine (impostor) distribution estimated from the training set. Let us also consider the first-order approximation of the integral used to compute the errors

$$P(s < s^*) = \frac{1}{2}\left(1 + \frac{2}{\sqrt{\pi}}\frac{s^* - \mu}{\sqrt{2}\sigma}\right) \tag{6}$$

Thus, substituting Equation (6) in Equation (4) we obtain:

$$D_j = |FRR(s_j) - FAR(s_j)| =$$

$$= |P(s < s_j|\text{genuine}) - P(s > s_j|\text{impostor})| =$$

$$= |P(s < s_j|\text{genuine}) - [1 - P(s < s_j|\text{impostor})]| =$$

$$= \left|\frac{1}{2}\left(1 + \frac{2}{\sqrt{\pi}}\frac{s_j - \mu_G}{\sqrt{2}\sigma_G}\right) - 1 + \frac{1}{2}\left(1 + \frac{2}{\sqrt{\pi}}\frac{s_j - \mu_I}{\sqrt{2}\sigma_I}\right)\right| =$$

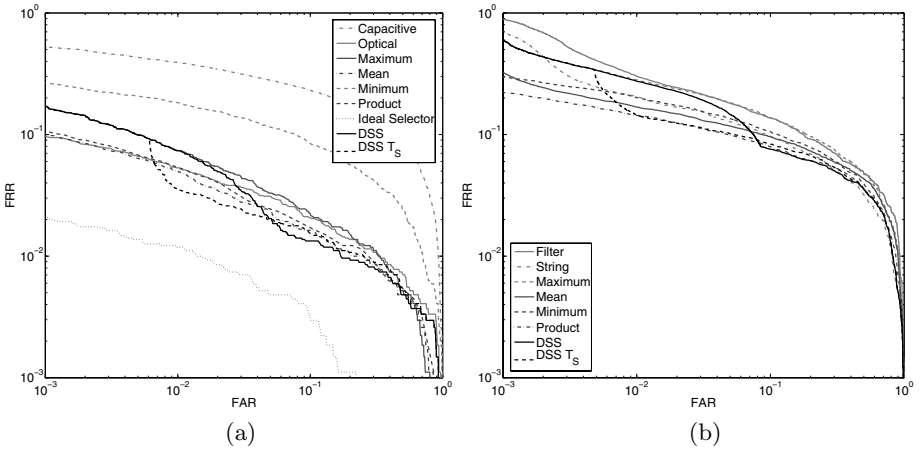
$$= \frac{1}{\sqrt{2\pi}}\left|\frac{s_j - \mu_G}{\sigma_G} + \frac{s_j - \mu_I}{\sigma_I}\right|$$

3.2 *Dynamic Score Selection* with Selection Threshold

It is easy to see that if the value of D_j are close to zero, i.e. the scores are close to the Equal Error Rate (EER) point, then the above estimation of the most likely state of nature is not reliable. On the other hand, it could be a safe decision to consider the input pattern as an impostor, and produce the output score using the *Min* rule. To this end, if the maximum value of D_j is smaller than some predefined threshold, than the *Min* rule is used.

4 Experimental Results

We tested the dynamic score selection technique on a fingerprint multisensor database that has been developed at our department [9]. This database is divided into two parts, each one containing 1200 fingerprints related to an optical sensor (Biometrika FX2000), and a capacitive sensor (Precise Biometrics MC100), respectively. These fingerprints have been taken from 20 volunteers. For each volunteer, ten impressions of three different fingers of both hands were acquired. We performed two sets of experiments, namely multi-sensor and multi-algorithmic. *Experiment 1* was performed using the "String" algorithm as minutiae matcher for both the available sensors. *Experiment 2* was performed using the "String"



(a) (b)

Fig. 2. R.O.C. curves for *Experiment 1* (a) and *Experiment 2* (b)

Table 1. Error rates of individual and combined systems in *Experiment 1*

Method	EER	1% FAR	1% FRR
Capacitive	18.65 %	39.15 %	92.67 %
Optical	3.52 %	5.33 %	33.44 %
Maximum	3.88 %	7.37 %	33.66 %
Mean	2.86 %	5.00 %	25.43 %
Minimum	9.14 %	18.33 %	71.03 %
Product	3.03 %	5.15 %	28.44 %
Ideal Selector	1.13 %	1.19%	1.52%
DSS	3.26 %	7.37 %	17.01 %
DSS T_S	2.57 %	3.52 %	25.41 %

Table 2. Error rates of individual and combined systems in *Experiment 2*

Method	EER	1% FAR	1% FRR
Filter	12.45 %	28.11 %	90.45 %
String	10.31 %	20.00 %	82.75 %
Maximum	12.45 %	28.15 %	90.47 %
Mean	9.67 %	16.87 %	89.94 %
Minimum	10.31 %	20.08 %	82.56 %
Product	8.44 %	14.36 %	86.45 %
Ideal Selector	0.04 %	0.02 %	0.00 %
DSS	8.31 %	27.45 %	82.04 %
DSS T_S	8.67 %	14.53 %	82.04 %

and "Filter" algorithms as minutiae matcher, and the capacitive sensor. The proposed DSS algorithm has been compared to four "fixed" fusion rules (Maximum, Mean, Minimum and Product). Performances were assessed in terms of: Equal Error Rate (*EER*), corresponding to the error rate computed for $FRR = FAR$; *1% FAR*, which is the FRR when FAR is fixed to 1%; *1% FRR*, which is the FAR when FRR is fixed to 1%.

Results in Tables 1 and 2 clearly show that the *ideal selector* exhibits a very small error rate. In addition, the related ROC curve reported in Figure 2a is below any other curve. It is worth noting that in Figure 2b the ROC curve of the ideal selector is barely visible, as it lies on the FAR axis, below all the other curves. This result shows that the proposed definition of ideal selection allows attaining very small error rates. It is worth noting that the score distributions of the ideal selector reported in Figure 1 are related to results of *Experiment 2*.

The analysis of the results in Table 1 shows that the *DSS* algorithm has the best single *1% FRR* error, while the *DSS* with selection threshold has the best

single error for *EER* and *1% FAR*. Thus it can be concluded that the DSS with selection threshold exhibits the best performances compared to other methods used in *Experiment 1*. The analysis of the results in Table 2 shows that the *DSS* algorithm exhibits the best single *1% FRR* and *EER* values. If the *DSS* with selection threshold is considered, while it exhibits the same value of *1% FRR* as the *DSS*, it also exhibits a lower value of *1% FAR*. Thus, by setting a threshold on the D_j value, it is possible to improve the performances in terms of *1% FAR*.

5 Conclusions

In this paper the basis for score selection was presented. A definition of *ideal selector* was given for score-level fusion, and an algorithm implementing the concepts of the ideal selector has been proposed. The experimental results, showed that score selection is an alternative to fusion strategies. While the concepts and the techniques illustrated in the paper were used for a fingerprint multi-sensor and multi-algorithmic system, they can be used for any score-based classification system and, in particular, with any kind of multi-modal biometric systems.

References

1. J. Bigun, J. Fierrez-Aguilar, J. Ortega-Garcia and J. Gonzales-Rodriguez, *Multimodal Biometric Authentication using Quality Signals in Mobile Communications*, Proc. IAPR ICIAP, IEEE CS Press, pp. 2–13, Mantova, Italy, Sept. 2003
2. G. Giacinto and F. Roli, *Dynamic Classifier Selection*, Proc. of the First Int. Work. on MCS, 2000, Lecture Notes In Computer Science 1857, Springer, pp. 177–189
3. V. Gunes, M. Ménard, P. Loonis and S. Petit-Renaud, *Combination, Cooperation and Selection of Classifiers: A State of the Art*, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 17, No. 8 (2003), pp. 1303–1324
4. L. Hong, A. K. Jain and S. Pankanti, *Can Multibiometrics Improve Performance?*, Proceedings AutoID'99, Summit, NJ, pp. 59–64
5. A.K. Jain, L. Hong and Y. Kulkarni, *A Multimodal Biometric System using Fingerprint, Face and Speech*, Proc. 2nd Int. Conf. on AVBPA, Mar. 1999, pp. 182–187
6. A.K. Jain, S. Prabhakar and A. Ross, *Fingerprint Matching:Data Acquisition and Performance Evaluation*, MSU Technical Report TR99–14, 1999
7. L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, John Wiley & Sons Inc., 2004, Chapter 6, pp. 189–202
8. G.L. Marcialis, F. Roli and P. Loddo, *Fusion of Multiple Matchers for Fingerprint Verification*, Proc. of the Work. on Machine Vision and Perception, Sept. 2002
9. G.L. Marcialis and F. Roli, *Fingerprint Verification by Fusion of Optical and Capacitive Sensors*, Pattern Recognition Letters, Vol. 25 (11), 2004, pp. 1315–1322
10. S. Prabhakar and A. K. Jain, *Decision-level Fusion in Biometric Verification*, Pattern Recognition, Vol. 35 (4), 2002, pp. 861–874
11. A.Ross, A. Jain and J.Z. Qian, *Information fusion in Biometrics*, Proc. International Conference on Image Processing, Rochester, New York, Sept. 22–25, 2002
12. K. Woods, W.P. Kegelmeyer and K. Bowyer, *Combination of Multiple Classifiers Using Local Accuracy Estimates*, IEEE Trans. PAMI, vol. 18 (4), 1997, pp. 405–410

Robust Face Recognition Based on Part-Based Localized Basis Images

Jongsun Kim and Juneho Yi

School of Information & Communication Engineering,
Sungkyunkwan University, Korea
Biometrics Engineering Research Center
{jskim, jhyi}@ece.skku.ac.kr

Abstract. In order for a subspace projection based method to be robust to local distortion and partial occlusion, the basis images generated by the method should exhibit a part-based local representation. We propose an effective part-based local representation method using ICA architecture I basis images that is robust to local distortion and partial occlusion. The proposed representation only employs locally salient information from important facial parts in order to maximize the benefit of applying the idea of "recognition by parts." We have contrasted our representation with other part-based representations such as LNMF (Localized Non-negative Matrix Factorization) and LFA (Local Feature Analysis). Experimental results show that our representation performs better than PCA, ICA architecture Ⅰ, ICA architecture Ⅱ, LFA, and LNMF methods, especially in the cases of partial occlusions and local distortions.

1 Introduction

Subspace projection techniques represent a face as a linear combination of low rank basis images. They employ feature vectors consisting of coefficients that are obtained by simply projecting facial images onto their basis images. Therefore, the performance of face recognition methods using subspace projection is directly related to the characteristics of their basis images. Among popularly used techniques are PCA, ICA and FLD [1-5]. ICA can be applied to face recognition in two different representations: ICA architecture I and II [2]. The goal of this research is to develop an optimal set of basis images for face recognition robust to partial occlusions and local distortions.

Facial image representations based on different basis images are illustrated in Figure 1. PCA and ICA architecture II basis images, as shown in Figure 1 (a) and (b), respectively, display global properties in the sense that they assign significant weights to potentially all the pixels. This accords with the fact that PCA basis images are just scaled versions of global Fourier filters [12]. In contrast, ICA architecture I basis images are spatially more localized. This local property of ICA architecture I basis images makes the performance of ICA architecture I based recognition methods robust to partial occlusion and local distortion, such as local changes in facial expression, because spatially local features only influence small parts of facial

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1026 – 1034, 2005.

images. However, ICA architecture I basis images do not display perfectly local characteristics, in the sense that pixels that do not belong to locally salient feature regions still have some nonzero weight values. These pixel values in non-salient regions would appear as noise and contribute to the degradation of the recognition.

In order for a subspace projection based method to be robust to partial occlusions and local distortions, its basis images should effectively realize a part-based local representation. Local representation is essential to robustness to partial occlusions and local distortions because successful face recognition can be achieved by representing some important facial parts that correspond to feature regions such as eyes, eye brows, nose and lips. This "recognition by parts" paradigm [11] has been popular in the object recognition research because the approach can be successfully applied to the problem of object recognition with occlusion. Among representative part-based local representations are Local Feature Analysis (LFA) [6] and Local Non-negative Matrix Factorization (LNMF) [13] methods. The LFA method extracts local features based on second-order statistics. However, basis image from the LFA representation are not perfectly localized as shown in Figure 1 (f). Thus, pixels in non-salient regions degrade the recognition performance in the case of partial occlusions and local distortions. Recently, the LNMF method was reported in the literature, which led to an improved version of Non-negative Matrix Factorization (NMF) [11]. In the LNMF method, locality constraints are imposed on the factorized matrices from NMF in terms of sparsity in matrix components. They successfully localized the components in basis images. However, the locality constraints do not guarantee that meaningful facial features should be localized in their basis images. As an example of the LNMF representation in Figure 1 (e) illustrates, some LNMF basis images represent regions such as cheek, forehead and jaw that are not discriminant features for face recognition.

We propose new basis images based on ICA architecture Ⅰ, called LS-ICA (locally salient ICA) basis images, where only locally salient feature regions are retained. The idea of "recognition by parts" can be effectively realized for face recognition using LS-ICA basis images since each LS-ICA basis image represent only locally salient regions. These regions correspond to important facial feature regions such as eyes, eye brow, nose and lips. Note that ICA architecture Ⅰ produces basis images that are localized edge filters [12], and they correspond to meaningful facial feature regions.

Our method for face recognition is characterized by two ideas: The first is the creation of the LS-ICA basis images using a modified version of Kurtosis maximization to remove residual nonlocal modulation in ICA architecture I basis images; these are used to represent faces. The second idea is to use LS-ICA basis images in the decreasing order of class separability so as to maximize the recognition performance. Experimental results show that LS-ICA performs better than PCA, ICA architecture Ⅰ, ICA architecture Ⅱ, LFA, and LNMF, especially in the cases of partial occlusions and local distortions such as local changes in facial expression.

The rest of this paper is organized as follows. Section 2 briefly describes the ICA, LFA and LNMF methods that are most relevant to our research. We present the proposed LS-ICA method in section 3. Section 4 gives experimental results.

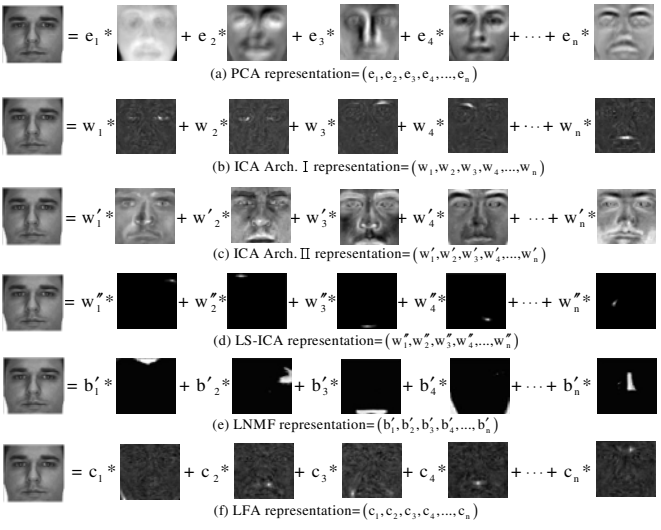


Fig. 1. Facial image representations using (a) PCA, (b) ICA architecture I, (c) ICA architecture II, (d) proposed LS-ICA, (e) LNMF and (f) LFA basis images: A face is represented as a linear combination of basis images. The basis images were computed from a set of images randomly selected from the AR database. Using LS-ICA basis images, the concept of "recognition by parts" can be effectively implemented for face recognition.

2 Related Work

2.1 ICA (Independent Component Analysis)

ICA is a widely used subspace projection technique that projects data from a high-dimensional space to a lower-dimensional space [2-4]. This technique is a generalization of PCA that decorrelates the high-order statistics in addition to the second-order moments. In this research, we compute ICA basis images using the FastICA algorithm [3] while other methods such as InfoMax [2] or Maximum likelihood [4] can also be employed.

The FastICA method computes independent components by maximizing non-Gaussianity of whitened data distribution using a kurtosis maximization process. The kurtosis measures the non-Gaussianity and the sparseness of the face representations [12]. The FastICA algorithm is briefly described as follows. Let $\mathbf{S}$ be the vectors of unknown source signals and $\mathbf{X}$ be vectors of observed mixtures. If $\mathbf{A}$ is an unknown mixing matrix, then the mixing model can be written as $\mathbf{X} = \mathbf{AS}$. The task is to estimate the independent source signals $\mathbf{U}$ by computing the separating matrix $\mathbf{W}$ that corresponds to the mixing matrix $\mathbf{A}$ using the following relation

$$\mathbf{U} = \mathbf{WX} = \mathbf{WAS}. \tag{1}$$

First, the observed samples are whitened. Let us denote the whitened samples by $\mathbf{Z}$. Then, we search for the matrix such that the linear projection of the whitened samples by the matrix $\mathbf{W}$ has maximum non-Gaussianity of data distribution. The kurtosis of $\mathbf{U}_i = \mathbf{W}_i^T \mathbf{Z}$ is computed as in equation (2) and the separating vector $\mathbf{W}_i$ is obtained by maximizing the kurtosis [3].

$$kurt(\mathbf{U}_i) = \left| E\left\{(\mathbf{U}_i)^4\right\} - 3\left(E\left\{(\mathbf{U}_i)^2\right\}\right)^2 \right| . \quad (2)$$

ICA can be applied to face recognition in two different architectures [2]. The ICA architecture I considers the input face images, $\mathbf{X}$, as a linear combination of statistically independent basis images, $\mathbf{S}$, combined by an unknown matrix, $\mathbf{A}$. On the other hand, the ICA architecture II finds statistically independent coefficients that represent input images. The coefficients obtained by projecting input images onto the statistically independent basis images are not statistically independent. The ICA architecture II basis images display global properties as shown in Figure 1 (c). Since the kurtosis maximization yields sparseness of basis images, the ICA architecture I basis images are spatially localized edge filters [12]. However, they do not display perfectly local characteristics in the sense that pixels that do not belong to locally salient feature regions still have some nonzero weight values. These pixel values would contribute to the degradation of the recognition performance in the case of local distortion and partial occlusion.

2.2 LFA (Local Feature Analysis)

LFA defines a set of local topographic kernels that are derived from the principal component eigenvectors $\mathbf{E}$ and coefficients $\mathbf{D}$ according to covariance matrix $\mathbf{S}$ using the following equation.

$$\mathbf{K} = \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T \text{ where } \mathbf{D}^{-\frac{1}{2}} = diag\left(\frac{1}{\sqrt{\lambda_i}}\right) \; i = 1,..., p \quad (3)$$

λ_i's are eigenvalues of the covariance matrix, $\mathbf{S}$. The rows of $\mathbf{K}$ contain the kernels. The kernels are topographic in that they are indexed spatially. The number of kernels corresponds to the number of pixels in an image. The LFA uses a sparsification algorithm in order to reduce the dimensionality of the representation. The algorithm iteratively selects kernels that have the largest mean reconstruction error. We are concerned with basis images from the LFA method. They not perfectly localized and pixels in non-salient regions contribute to the degradation of the recognition performance.

2.3 LNMF (Local Non-negative Matrix Factorization)

The LNMF method [13] is a technique that improves the standard NMF method [11]. It is aimed at learning spatially localized, parts-based subspace representation of basis images by imposing additional constraints on the NMF basis. The following objective function is used to compute LNMF basis image:

$$D(X \,\|\, BH) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(X_{ij}\log\frac{X_{ij}}{[BH]_{ij}} - X_{ij} + [BH]_{ij} + \alpha U_{ij} \right) - \beta\sum_i V_{ii} \quad (4)$$

where $\alpha, \beta > 0$ are some constants, $U = B^T B$, $V = HH^T$ and $B, H \geq 0$ means that all entries of basis images B and coefficients H are non-negative. The minimization of $U = B^T B$ imposes both maximum sparsity in H and minimum redundancy between

different bases. On the other hand, by maximizing $\sum_i V_{ii}$, basis components that carry much information about the training images are retained. Refer to [13] for further justification of the objective function. The LNMF update rule for H uses square root as in equation (5) to satisfy the additional constraints. The update for B is identical to that of NMF [14].

$$H_{aj}^{(t+1)} = \sqrt{H_{aj}^{(t)} \sum_i B_{ai}^{T(t)} \frac{X_{ij}}{(B^{(t)}H^{(t)})_{ij}}} \tag{5}$$

Figure 1 (e) shows an example of the LNMF representation. As described earlier, the additional constraints only focus on locality and it is not necessarily guaranteed that meaningful facial features are localized in their basis images. We can see that some LNMF basis images represent regions such as cheek, forehead and jaw that are not discriminant features for face recognition.

3 The LS-ICA (Locally Salient ICA) Method

The computation of LS-ICA basis images consists of two steps: The first step is concerned with the creation of part-based local basis images based on ICA architecture I. The second is to order the basis images obtained in the order of class separability for good recognition performance.

The LS-ICA method creates part-based local basis images by imposing additional localization constraint in the process of the kurtosis maximization. The solution at each iteration step is weighted so that it becomes sparser by only emphasizing large pixel values. Let $\mathbf{u}$ be a solution vector at an iteration step, we can define a weighted solution as $\mathbf{b}$ where $\mathrm{b}_i = |\mathrm{u}_i|^{\alpha} \mathrm{u}_i$ and $\mathbf{b} = \mathbf{b}/\|\mathbf{b}\|$. $\alpha > 1$ is a small constant. The kurtosis is maximized in terms of $\mathbf{b}$ instead of $\mathbf{u}$ as in equation (6).

$$kurt(\mathbf{b}) = \left| E\left\{(\mathbf{b})^4\right\} - 3\left(E\left\{(\mathbf{b})^2\right\}\right)^2 \right| . \tag{6}$$

A solution to the above function can be found by using the following update rules:

$$\mathbf{w}^{(t+1)} = E\left\{ |\mathrm{u}_i|^{\alpha} \mathbf{Z} \left(|\mathrm{u}_i|^{\alpha} \mathbf{w}^{T(t)} \mathbf{Z} \right)^3 \right\} \tag{7}$$

where $\mathbf{w}$ is a separating vector and $\mathbf{Z}$ contains whitened image samples. The resulting basis image is $\mathrm{b}_i = |\mathrm{u}_i|^{\alpha} \left(\mathbf{w}^T \mathbf{Z}\right)_i$. As an alternative method, we would like to point out that other simple operations such as morphological operations can be employed to detect salient regions from ICA architecture I basis images.

We then compute a measure of class separability, r, for each LS-ICA basis vector and sort the LS-ICA basis vectors in the decreasing order of class separability [2]. The class separability, r, of a basis vector is defined as the ratio

$$r = \frac{\sigma_{between}}{\sigma_{within}} . \tag{8}$$

$\sigma_{between}$ and σ_{within} denote, respectively, the between-class variability and within-class variability of its corresponding projection coefficients of training images We then create new LS-ICA basis images from the LS-ICA basis images selected in the decreasing order of the class separability. This way, we can achieve both dimensionality reduction and good recognition performance. The LS-ICA representation is based on the idea of localized edge filters that come from ICA basis images. The resulting basis images contain localized facial features that are discriminant for face recognition.

4 Experimental Results

We have used facial image databases such as the AT&T [8] and AR [9] and databases in order to compare the recognition performance of LS-ICA with that of PCA, ICA architecture Ⅰ, ICA architecture Ⅱ, LFA and LNMF methods. Figure 2 shows example images from these databases. For fair comparisons with the above methods, their basis images were also used in the decreasing order of class separability, r. We have computed recognition performances for three different distance measures (L1, L2, cosine) to see if there is any performance variation depending on the distance measure used [7].

In order to show the performance comparisons under occlusion, we have used the AT&T database. It consists of 400 images of 40 persons, 10 images per person. The images are taken against a dark homogeneous background and the subjects are in an up-right, frontal position with tolerance for some side movement. A set of 10 images for each person is randomly partitioned into five training images and five testing images. The occlusion is simulated in an image by using a white patch of size $s \times s$ with $s \in \{10, 20, 30\}$ at a random location as shown in Figure 3. Figure 4 compares the six representations under varying degrees of occlusion, in terms of the recognition accuracies versus the size $s \times s$ of occluding patch. The LS-ICA and LNMF methods performed better than the other methods under partial occlusion, especially as the patch size increases. The LS-ICA method that only makes use of locally salient information from important facial parts achieved higher recognition rates than the LNMF method. The LFA and ICA architecture I methods appear influenced by pixels not belonging to salient regions. The ICA architecture Ⅰ method showed better performance than the LFA method. Bartlett et al. [15] have also reported that the ICA representation performs better than the LFA representation in cases of facial expression analysis and face recognition. This experimentally shows that the ICA architecture Ⅰ better represents some important facial parts than the LFA method.

The AR database contains 800 frontal facial images from 100 subjects. The number of images used for training and testing are 200 and 600 images, respectively. Test images contain local distortions and occlusions such as changes in facial expression

and sunglasses worn. Figures 5 shows the recognition performances for the AR database. The recognition rate of the LS-ICA method was consistently better than that of PCA, ICA architecture Ⅰ, ICA architecture Ⅱ, LNMF and LFA methods regardless of the distance measures used. We can see that the LS-ICA representation is an effective part-based local representation for face recognition robust to local distortion and partial occlusion.

Fig. 2. Example images from AT&T (left) and AR (right) facial databases

Fig. 3. Example AT&T images having random occluding patches of sizes (from left to right) 10x10, 20x20, and 30x30

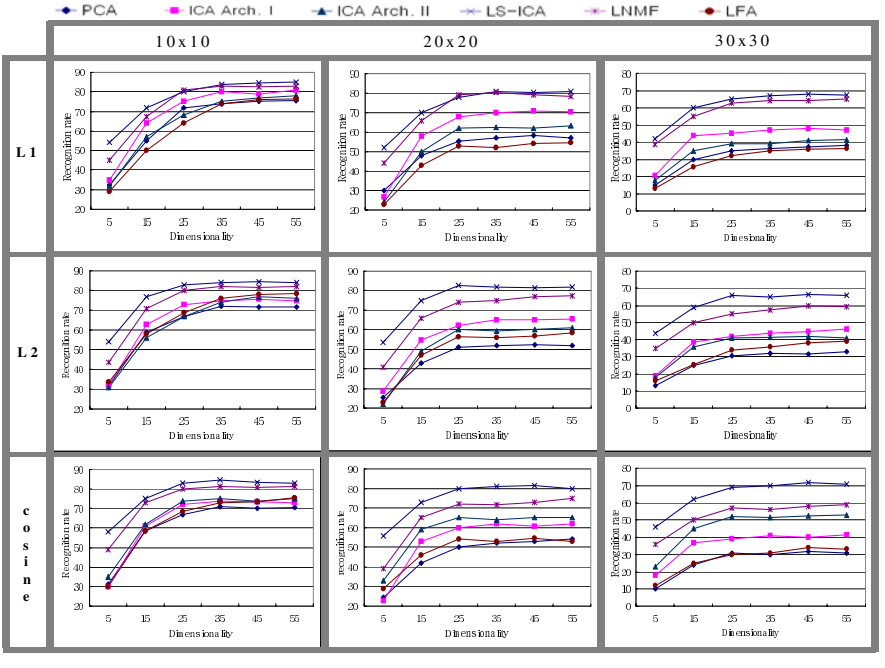


Fig. 4. Recognition performance versus the size (in 10x10, 20x20, 30x30) of occluding patches for PCA, ICA1, ICA2, LNMF, LFA, and the LS-ICA method for the AT&T database

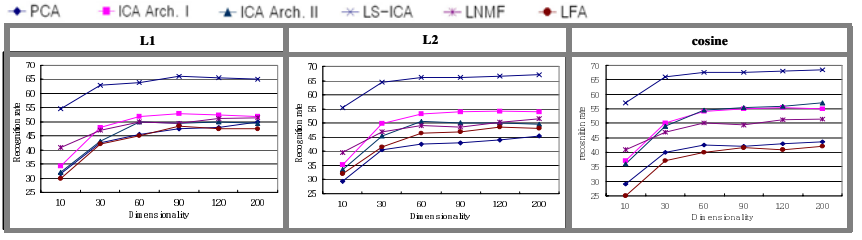


Fig. 5. The recognition performance of PCA, ICA1, ICA2, LNMF, LFA and LS-ICA methods for the AR facial databases

5 Conclusions

Application of the idea of "recognition by parts" is necessary for the problem of face recognition under partial occlusion and local distortion. To maximize the benefit of "recognition by parts", we have proposed an effective part-based local representation based on ICA architecture I basis images. The basis images are created by imposing additional localization constraint in the process of computing ICA architecture I basis images. The performance of the LS-ICA method was consistently better than other representative local representation based methods, especially in the cases of facial images that have partial occlusions and local distortions such as changes in facial expression.

Acknowledgement

This work was supported by the Korea Science Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei University and BK21.

References

[1] M. A. Turk and A. P. Pentland, "Eigenfaces for recognition," *Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[2] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face Recognition by Independent Component Analysis," *IEEE Trans. Neural Networks*, vol. 13, no. 6, pp. 1450-1464, 2002.

[3] Aapo Hyvarinen and Erki Oja, "Independent component analysis: a tutorial," http://www.cis.hut.fi/~aapo/papers/IJCNN99_tutorialweb/, 1999.

[4] A. Hyvärinen, "The Fixed-point Algorithm and Maximum Likelihood Estimation for Independent Component Analysis," *Neural Processing Letters*, vol. 10, pp. 1-5, 1999.

[5] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE PAMI*, vol. 19, no. 7, pp. 711-720, 1999.

[6] P. Penev and J. Atick, "Local Feature Analysis: A general statistical theory for object representation," *Network: Computation in Neural Systems*, vol. 7, no. 3, pp. 477-500, 1996.
[7] B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge, "Recognizing faces with PCA and ICA," *Computer Vision and Image Understanding*, vol. 91, no. 1, pp. 115-137, 2003.
[8] http://www.uk.research.att.com/facedatabase.html.
[9] A. M. Martinez and R. Benavente, "The AR face database," CVC Tech, 1998.
[10] A. P. Pentland, "Recognition by parts," *IEEE Proceedings of the First International Conference on Computer Vision*, pp. 612-620, 1987.
[11] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788-791, 1999.
[12] A. J. Bell and T. J. Sejnowski, "The Independent Components of Natural Scenes are Edge Filters," *Vision Research*, vol. 37, no. 23, pp. 3327-3338, 1997.
[13] S. Z. Li, X. W. Hou, H. J. Zhang, "Learning Spatially Localized, Parts-Based Representation," *Computer Vision and Pattern Recognition*, vol. 1, pp. 207-212, 2001.
[14] S. Wild, J. Curry, A. Dougherty, "Motivating Non-Negative Matrix Factorizations," *In Proceedings of the Eighth SIAM Conference on Applied Linear Algebra*, July 2003.
[15] M. S. Bartlett, *Face Image Analysis by Unsupervised Learning,* Kluwer Academic Publishers, 2001.

Combining Multiple Matchers for Fingerprint Verification: A Case Study in FVC2004

J. Fierrez-Aguilar[1,⋆], Loris Nanni[2], J. Ortega-Garcia[1], Raffaele Cappelli[2], and Davide Maltoni[2]

[1] ATVS, Escuela Politecnica Superior, Universidad Autonoma de Madrid, Avda. Francisco Tomas y Valiente 11, Campus de Cantoblanco, 28049 Madrid, Spain
{julian.fierrez, javier.ortega}@uam.es
[2] Biometric Systems Lab., DEIS, Universita di Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
{lnanni, rcappelli, dmaltoni}@deis.unibo.it

Abstract. Combining different algorithms submitted to the Third International Fingerprint Verification Competition (FVC2004) is studied. For this work, the matching results of more than 40 fingerprint systems from both academy and industry are available on standard benchmark data. In particular, we concentrate on score-level fusion of the different algorithms, studying the correlation between them by using feature-subset-selection techniques. Based on the general algorithmic descriptions provided by the participants, some interesting experimental findings are obtained.

1 Introduction

The increasing interest in a wide variety of practical applications for automatic personal identification and authentication has resulted in the popularity of biometric recognition systems [1]. In particular, the use of fingerprint images has initiated much research, with a large number of different algorithmic approaches proposed during the last decades [2]. As a result, recent efforts have been conducted in order to establish common evaluation scenarios enabling a fair comparison between competing systems [3]. In the case of fingerprint recognition, a series of International Fingerprint Verification Competitions (FVC) [4] have received great attention both from the academy and the industry. These competitions have provided common data and procedures widely available now for further research [2]. Other recent comparative benchmark studies include the Fingerprint Vendor Technology Evaluations organized by NIST [5].

On the other hand, one recent focus of interest in biometrics research is the successful combination of different sources of information resulting in the so-called multi-biometric approaches [6]. The general scheme of the multiple

⋆ This work has been carried out while J. F.-A. was on a research visit at Bologna University.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1035–1042, 2005.

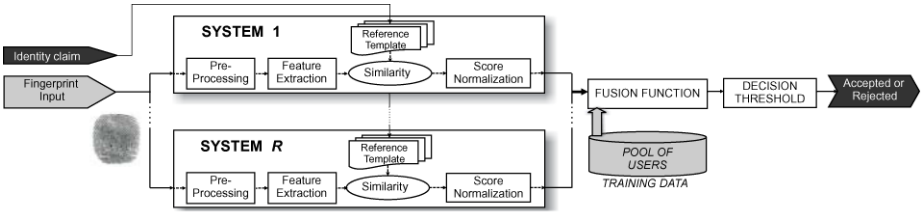


Fig. 1. System model of score-level multiple matcher approach for fingerprint verification

matcher approach that will be considered in the rest of the paper is depicted in Fig. 1 for the case of fingerprint verification.

The aim of this work is to study the combination of different fingerprint recognition systems submitted to FVC2004 and to analyze the benefits and limits of the resulting multiple matcher approaches. Based on the general algorithmic descriptions provided by most of the participants, and the experimental evidence obtained, we finally draw a number of conclusions of practical interest.

2 FVC2004: Fingerprint Verification Competition

Details of the competition and results were presented in [4] and more information appears in [7]. Two different sub-competitions (*open* and *light*) were organized. Light category was intended for algorithms with restricted computing and memory usage, 26 algorithms participated in this case. 41 algorithms participated in the open category. In Table 1 we include general algorithmic descriptions of most of the systems, as provided by the participants, following the taxonomy in [2]. Pointers to the identities of non-anonymous participants, individual results, and comparative charts can be found at [7]. In this work, we focus on combining the algorithms competing in the *open* category.

Data for the competition consists of 4 different databases, the first three acquired with different sensors and the last of them created with a synthetic

Fig. 2. Fingerprint examples from the four databases used in FVC2004 (left to right): DB1 (CrossMatch V300), DB2 (Digital Persona UareU 4000), DB3 (Atmel FingerChip), and DB4 (SFinGe v3.0).

Table 1. High-level description of systems from 29 participants in FVC2004. Participant *071*: only performs segmentation in the light category, alignment type is Displacement+Rotation in the light category and Non-linear in the open, raw Image parts and Correlation are used only in the open category. Participant *101*: Segmentation is performed only on DB1.

	Preprocessing		Alignment		Features								Matching				
Participant	Segmentation	Enhancement	Before Matching, During matching	Displacement, Rotation, Scale, Non-linear	Minutiae	Singular points	Ridges	Ridge counts	Orientation field	Local ridge frequency	Texture measures	Image parts	Minutiae (global)	Minutiae (local)	Ridge pattern (geometry)	Ridge pattern (texture)	Correlation
002	√	√	D	NL	√				√				√	√			
009	√	√	BD	DRS	√	√	√	√	√	√			√				
016		√			√		√						√				√
026				DR	√			√	√				√				
027	√	√	D	DRS							√					√	√
039	√	√	D	N	√				√	√			√				
041	√	√	D	DR	√		√								√		
047	√		D	DRSN	√	√	√	√	√	√	√	√	√	√		√	
049	√	√	D	DR	√	√	√		√	√			√	√			
050	√	√	B	DR	√				√				√				
051	√	√	D	DR	√	√		√	√				√	√			√
067	√	√	D	DRN	√				√				√				
068	√	√	D	DR	√		√						√				√
071	(√)	√	D	DR(N)	√	√	√	√	√			(√)		√	√		(√)
072	√	√	D	DR	√	√			√				√				
075	√	√	B	DR	√									√			
078	√	√	D	DRS	√								√				
087	√	√	D	DR	√				√	√	√		√	√	√		
097	√		D	DR	√	√			√	√			√	√			
099	√	√	D	DRN	√								√				
101	(√)	√	BD	DRS		√	√	√	√	√		√			√		√
103	√	√			√				√					√			
104	√	√	D	DR	√									√			
105	√	√	B	DR	√	√			√					√	√		
106					√		√							√			
107		√	D	DRS	√	√							√	√			
108	√	√	D	DR	√	√			√					√			
111	√	√	D	DR	√				√				√				
113	√	√	D	N	√	√	√		√	√		√	√	√			√

generator, see Fig. 2 for example images. Worth noting, the image quality is low to medium due to exaggerated plastic distortions, and artificial dryness and moistness [4]. Each database comprises 100 different fingers with 8 impressions per finger. Different fingers were used for different databases.

The experimental protocol was as follows. The 8 impressions of each finger were matched against each other avoiding symmetric matches, thus generating $(100 \times 8 \times 7)/2 = 2800$ genuine matching scores. The first impression of each user was also matched against the first impression of every other user avoiding symmetric matches, thus generating $(100 \times 99)/2 = 4950$ impostor matching scores. The matching scores are similarity values ranging between 0 and 1. A number of performance indicators were then obtained including score histograms, verification error rates, computing time, memory allocated, and others [4]. In this work we carry out comparative experiments focusing on the Equal Error Rate between individual and combined algorithms.

3 Methods

3.1 Score Fusion Methods

Three different combination strategies are evaluated in the present work.

Sum. This basic fusion method consists in averaging the matching scores provided by the different systems. Under some mild statistical assumptions [8,9], this simple method is demonstrated to give good results for the biometric authentication problem.

SVM. Another popular approach to combining multiple matchers in biometric recognition is to treat the combination stage as a second-level pattern recognition problem on the matching scores that are to be fused [10], and then use standard learning paradigms in order to obtain combining functions. Comparative studies exist in this regard [11], favoring the Support Vector Machine-based fusion approach over others [12]. Linear and RBF kernels are used in the present work [13].

Dempster-Shafer. Another group of methods well adapted to multiple classifier approaches is based on evidence accumulation [14]. From this group, we have selected and implemented the fusion scheme based on Dempster-Shafer theory as described in [14].

3.2 Feature Subset Selection Method

In order to study the effects of the correlation between different systems on the performance of their combination, we have considered the fusion stage as a second-level 2-class pattern recognition problem. The similarity score output of each system is seen as a different feature, and the two classes correspond to impostor and genuine attempts, respectively.

Best fingerprint systems to combine are then chosen by running Sequential Forward Floating Selection (SFFS) [15] on the similarity scores available from

the competition. Goodness of the feature subset was defined as the empirical EER obtained using the sum rule.

4 Experiments

4.1 Experimental Protocol

Feature selection and fusion experiments are carried out separately for the 4 different databases considered. Whenever training was needed for the fusion schemes (i.e., SVM and Dempster-Shafer), 2-fold cross-validation was used, dividing the available scores in two partitions (different fingers in each of them).

4.2 Results

A comparative chart is depicted in Fig. 3 for the case of combining 2, 4, and 6 fingerprint systems (horizontal axis of each subplot) with all the fusion strategies considered (bars with different greyscale), on the 4 databases considered

Table 2. Systems chosen with SFFS. EER in %

DB1				DB2				DB3				DB4			
Participant	Ranking on DB1 (EER)	EER on DB1	EER on DB1 (Sum)	*Participant*	Ranking on DB2 (EER)	EER on DB2	EER on DB2 (Sum)	*Participant*	Ranking on DB3 (EER)	EER on DB3	EER on DB3 (Sum)	*Participant*	Ranking on DB4 (EER)	EER on DB4	EER on DB4 (Sum)
047	1st	1.97		*039*	1st	1.58		*047*	1st	1.18		*071*	1st	0.61	
047	1st	1.97	1.45	*039*	1st	1.58	0.92	*101*	2nd	1.20	0.28	*071*	1st	0.61	0.48
101	2nd	2.72		*101*	7th	3.56		*075*	5th	1.85		*101*	2nd	0.80	
047	1st	1.97	1.20	*039*	1st	1.58	0.73	*101*	2nd	1.20	0.23	*071*	1st	0.61	0.39
101	2nd	2.72		*101*	7th	3.56		*075*	5th	1.85		*101*	2nd	0.80	
004	6th	4.10		*103*	14th	4.99		*078*	29th	7.56		*113*	12th	1.98	
047	1st	1.97	1.17	*039*	1st	1.58	0.67	*101*	2nd	1.20	0.21	*071*	1st	0.61	0.31
101	2nd	2.72		*004*	3rd	2.79		*075*	5th	1.85		*101*	2nd	0.80	
004	6th	4.10		*101*	7th	3.56		*004*	6th	1.89		*039*	4th	1.07	
052	19th	8.41		*103*	14th	4.99		*002*	13th	3.82		*075*	31th	5.99	
047	1st	1.97	1.03	*039*	1st	1.58	0.61	*047*	1st	1.18	0.21	*071*	1st	0.61	0.26
101	2nd	2.72		*004*	3rd	2.79		*101*	2nd	1.20		*101*	2nd	0.80	
097	3rd	3.38		*113*	4th	3.17		*039*	4th	1.78		*039*	4th	1.07	
049	5th	3.91		*101*	7th	3.56		*075*	5th	1.85		*027*	18th	3.22	
004	6th	4.10		*103*	14th	4.99		*097*	16th	4.16		*075*	31th	5.99	
047	1st	1.97	1.02	*039*	1st	1.58	0.61	*047*	1st	1.18	0.23	*071*	1st	0.61	0.25
101	2nd	2.72		*004*	3rd	2.79		*101*	2nd	1.20		*101*	2nd	0.80	
097	3rd	3.38		*113*	4th	3.17		*039*	4th	1.78		*039*	4th	1.07	
049	5th	3.91		*101*	7th	3.56		*075*	5th	1.85		*047*	10th	1.76	
004	6th	4.10		*048*	12th	4.67		*009*	12th	3.74		*027*	18th	3.22	
048	14th	7.47		*103*	14th	4.99		*097*	16th	4.16		*075*	31th	5.99	

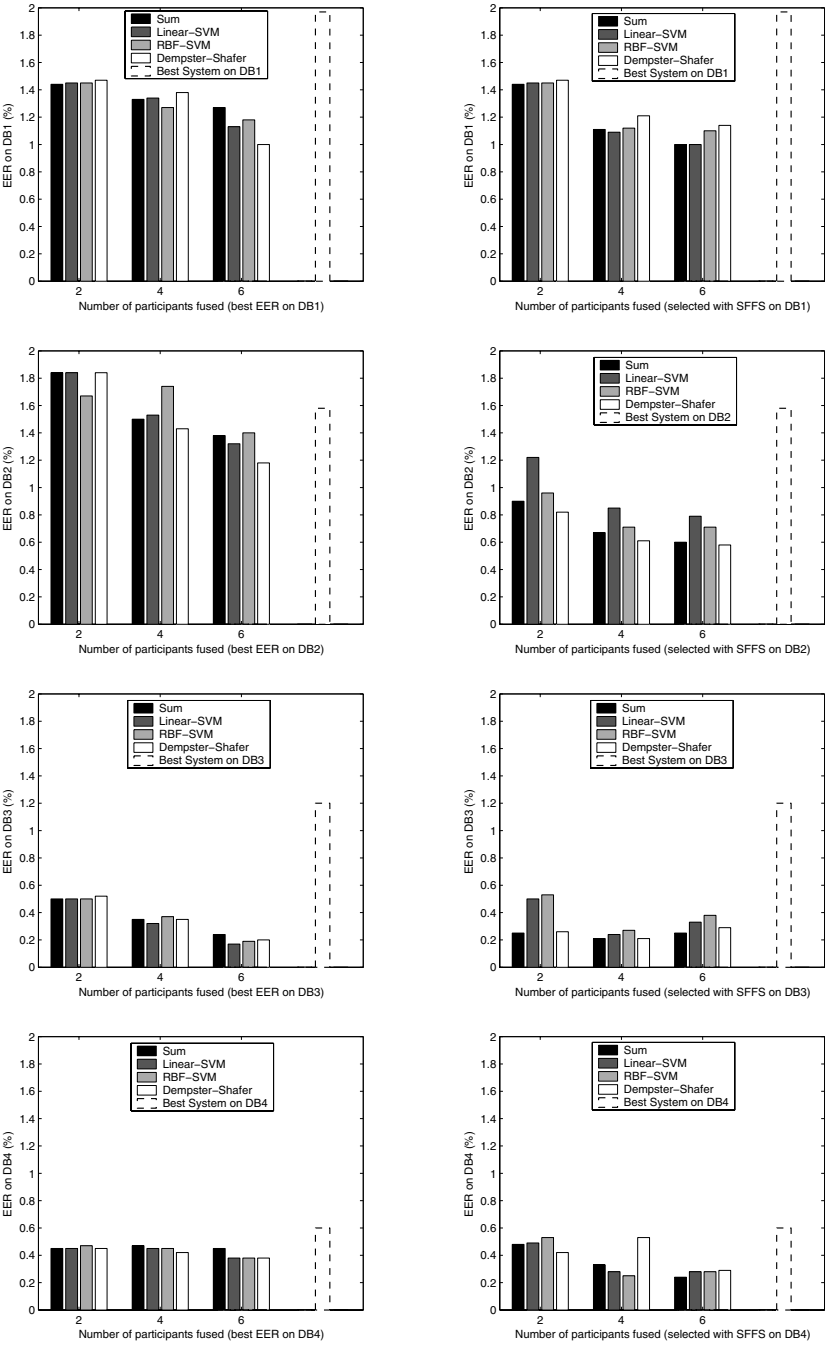


Fig. 3. Results. Open Category.

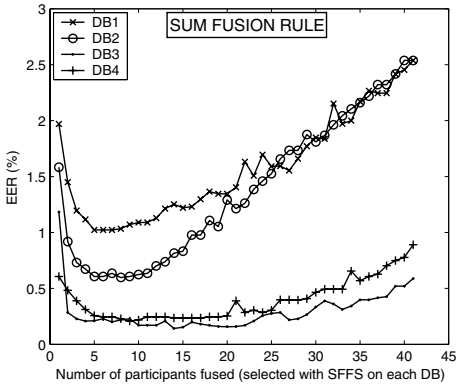


Fig. 4. Verification performance results for an increasing number of fingerprint systems selected with SFFS for fusion

(different rows), either combining the best individual systems on each database (left column) or selecting the best combination through SFFS (right column). The performance of the best individual system in each database is also shown as a dashed bar for reference. Substantial performance improvements are obtained when combining different systems in most of the cases, specially DB3. Also interesting is the fact that combining systems chosen with SFFS always leads to improved performance and this is not the case when combining best individual systems (e.g., DB2). Also interestingly, no particular fusion method seems to consistently outperform the others.

In Fig. 4 we plot the results of the combination of an increasing number of systems selected with SFFS for the 4 databases considered. The performance improves with the fusion of up to 5 to 7 systems and then deteriorates when combining more than 10 systems. The systems automatically selected by the SFFS algorithm are shown in Table 2, following the system numbering of Table 1.

5 Discussion and Conclusions

A number of experimental findings can be extracted from the results. Firstly, the simple fusion approach based on sum rule is not clearly outperformed by more complex trained fusion approaches. Secondly, the combination of the top performing individual systems can be outperformed by other combinations (e.g., the best 2-system combination on DB3 is obtained by fusing the systems individually ranked as 2nd and 5th, see Table 2). Analyzing various individual cases, best combinations are usually obtained when combining systems that are based on heterogenous matching strategies, such as minutia-based with ridge- [16] and/or correlation-based [17].

Also worth noting, the maximum performance is found when combining a small number of systems (about 7 in this case study). Initially, the performance

improvement is significantly high, but as we increase the number of systems to fuse the performance stabilizes reaching a maximum. If more systems are combined, the performance actually degrades.

Acknowledgements

This work has been supported by Spanish TIC2003-08382-C05-01, Italian 2004098034, and European IST-2002-507634 projects. Authors also thank Prof. Anil K. Jain for helpful comments. J. F.-A. is supported by a FPI scholarship from Comunidad de Madrid.

References

1. Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. IEEE Trans. on Circuits and Systems for Video Technology **14** (2004) 4–20
2. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer (2003)
3. Wayman, J., Jain, A., Maltoni, D., Maio, D., eds.: Biometric Systems: Technology, Design and Performance Evaluation. Springer (2005)
4. Maio, D., Maltoni, D., et al.: FVC2004: Third Fingerprint Verification Competition. In: Proc. ICBA, Springer LNCS-3072 (2004) 1–7
5. Wilson, C., et al.: FpVTE2003: Fingerprint Vendor Technology Evaluation 2003 (NISTIR 7123) website: http://fpvte.nist.gov/.
6. Jain, A.K., Ross, A.: Multibiometric systems. Communications of the ACM **47** (2004) 34–40
7. FVC2004 website: http://bias.csr.unibo.it/fvc2004.
8. Bigun, E.S., et al.: Expert conciliation for multi modal person authentication systems by Bayesian statistics. In: Proc. AVBPA, LNCS-1206 (1997) 291–300
9. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Trans. on Pattern Anal. and Machine Intell. **20** (1998) 226–239
10. Verlinde, P., Chollet, G., Acheroy, M.: Multi-modal identity verification using expert fusion. Information Fusion **1** (2000) 17–33
11. Ben-Yacoub, S., Abdeljaoued, Y., Mayoraz, E.: Fusion of face and speech data for person identity verification. IEEE Trans. on Neural Networks **10** (1999) 1065–1074
12. Gutschoven, B., Verlinde, P.: Multi-modal identity verification using support vector machines (SVM). In: Proc. FUSION, IEEE Press (2000) 3–8
13. Fierrez-Aguilar, J., et al.: Exploiting general knowledge in user-dependent fusion strategies for multimodal biometric verification. In: Proc. ICASSP (2004) 617–620
14. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion: An experimental comparison. Pattern Recogn. Lett. **34** (2001) 299–314
15. Jain, A., Zongker, D.: Feature selection: Evaluation, application, and small sample performance. IEEE Trans. on Pattern Anal. and Machine Intell. **19** (1997) 153–158
16. Ross, A., Jain, A.K., Reisman, J.: A hybrid fingerprint matcher. Pattern Recognition **36** (2003) 1661–1673
17. Nandakumar, K., Jain, A.K.: Local correlation-based fingerprint matching. In: Proc. Indian Conf. on Comp. Vision, Graphics and Image Process. (2004) 503–508

Classifier Combination for Face Localization in Color Images

Rachid Belaroussi, Lionel Prevost, and Maurice Milgram

LISIF Université Pierre et Marie Curie BC252,
4 place Jussieu 75252 Paris cedex 05 France
rachid.belaroussi@gmail.com
{lionel.prevost, maurice.milgram}@lis.jussieu.fr

Abstract. We present a new method dedicated to the localization of faces in color images. It combines a connexionist model (auto-associative network), an ellipse model based on Generalized Hough Transform, a skin color model and an eyes detector that results in two features. A linear combination of the 3 first models is performed to eliminate most of non face regions. A connexionist combination of the four detectors response is performed on the remaining candidates. Given an input image, we compute a kind of probability map on it with a sliding window. The face position is then determined as the location of the absolute maximum over this map. Improvement of baseline detectors localization rates is clearly shown and results are very encouraging.

1 Introduction

Face detection in an image without any hypothesis is a tough task because of the high variability of the pattern to be detected [1]. As in many detection issues, it is almost impossible to define the opposite class, the non-face patterns, which drives researchers to choose the model-based approach. Solutions implemented in a large number of face detection applications (biometric, presence detection, visiophony, indexation, car driver detection, virtual reality, lips reading) start with simplifying the problem by making assumptions : fixed camera and known background, use of motion information [2], strong hypothesis on the face location, special background for an easy extraction of the silhouette or special lighting conditions (use of infra-red, for example). Face localization (the face is in the image and we want to know where) is not simpler without additional assumption.

We find here the two approach common in Pattern Recognition : structural and global. Structural approaches try to detect primitives of the face (eyes, mouth, nose, head edge) then combine the results using geometrical and radio metrical models, or constellations analysis [3]. Global approaches process a sub-image of the input image into a feature vector (momentum, projection, gray level, wavelet...). These approaches estimate the classifier parameters on a training set. In the global approach, parameters can be weights (neural networks) [4] or terms of a covariance matrix (statistical classifier). A choice is then to be made between the model approach and the discriminative one. A model does not require counter examples, which may seems an

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1043 – 1050, 2005.

advantage but actually decreases classifier efficiency : generalization in a high dimension space (221 for 13x17 sub-images) is tough without knowing where are the vectors that might be confused. Another way is to design a combination of several detectors (classifiers). [5] did it to perform face detection and classifier combination has also been used in character [6], and face recognition [7].

In order to get the best of both worlds, our method makes co-operate holistic and structural approaches. Our face localization approach combines in a first stage (a pre-filter resulting in regions of interest) an auto-associator network appearance based model, and an ellipse detector both based on the image gradient's direction, and a coarse skin color model in YCbCr color space. A second stage integrates these three detectors and two features related to the eyes as shown in fig. 1.

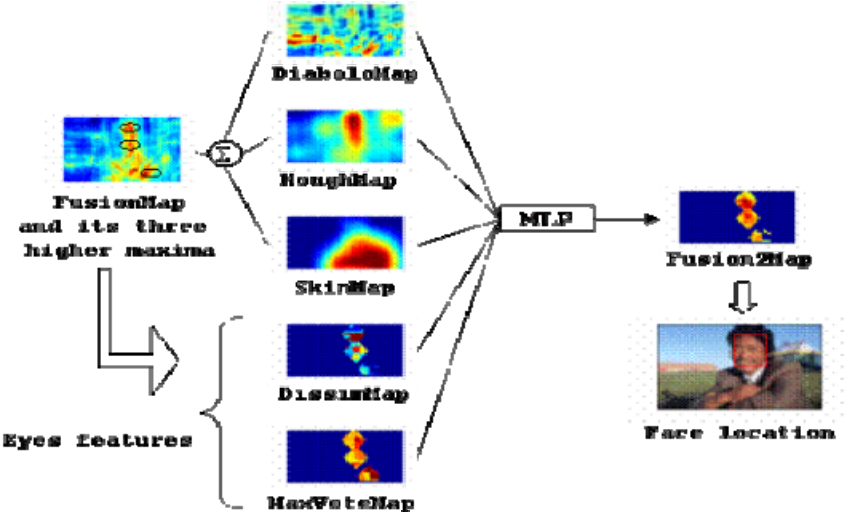


Fig. 1. Overview of the face localization system

Section 2 describes all these detectors, and their combination is detailed in Section 3. We present in Section 4 our experimental results and the benefits of the combination. The last section is devoted to conclusions and prospects.

2 Basic Detectors

2.1 Edge Orientation Evaluation

A part of the information of a face image lies in orientation of its edges. A huge advantage of the edge's orientation is its relative invariance to the skin tone. Three of the four detectors presented in the next sections use this information : the appearance-based model, the ellipse detector and the eyes detector.

Evaluation of the gradient orientation on edge image requires a low-pass filtering of the image. Gradient field estimation uses Roberts masks, so that gradient magnitude on the *x*-axis and *y*-axis are computed as follow:

$$I_x = I_{filtered} \otimes \begin{bmatrix} 1 & -1 \end{bmatrix} \text{ and } I_y = I_{filtered} \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Then, the gradient magnitude is thresholded. For the generalized Hough transform, a global threshold is applied over the whole input image. This threshold was optimized over 168 images and is equal to 12. The thresholding for the neural network is defined over each 13x17 sub-windows of the input image, so that 20% of the pixels are then regarded as edges. Orientations of these edge pixels are then quantized on N=36 values, except for the eyes detector which will be described in the last sub-section.

2.2 Appearance-Based Model: The Auto-associative Multi-layer Perceptron (Diabolo)

Pre-processing of training examples are described in [8]. Cosinus and sinus of the edge orientations inside an elliptical mask provide a 290 features vector for each face example. Processing of these examples is done with an auto-associator neural network, the so called “Diabolo” [9]. This network was successfully used for handwritten characters recognition [10], face detection [9] and compression [11]. It is trained to reconstruct an output identical to its input. It implements a specialized compression for its hidden layer has much less units than input or output does. So a non-face image should be badly compressed and the reconstruction error (square root of the mean square error between the input and the calculated output) would be higher than for a face image.

The neural network is trained using 1602 face images as a training set and training is stopped by cross-validation on 178 images. Weight and bias values are updated according to gradient descent with adaptive learning rate. After an exhaustive search we found that for 290 inputs (corresponding to a 17x13 retina) the optimal number of hidden cells is 18. The gray level image is scanned at the resolution corresponding to the size of the face with a 13x17 sliding retina, and at each position of the image a reconstruction error is calculated. An array of reconstruction errors is calculated, we will referre to it as DiaboloMap.

2.3 Ellipse Detector Based on Generalized Hough Transform

Orientation of the gradient over the whole gray level image is then determined. Then, a Generalized Hough Transform (GHT) is performed on the resulting orientation map. Faces are modeled as vertical ellipses with a specific eccentricity so we can build up a lookup table to cast votes from each edge pixel, knowing its gradient orientation [8]. It provides a vote array which maximum correspond in the image to the position most likely to be the center of an ellipse with a horizontal minor axis a=8, and a vertical major axis b=10. The vote map is scanned using a 13x17 mexican hat which provides a new score map used to defined face location, we will referred to it as HoughMap (see fig. 2).

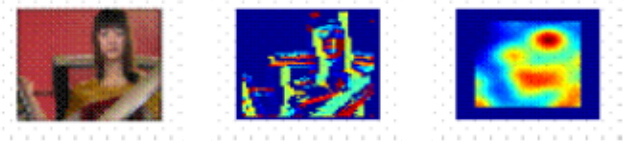


Fig. 2. Original image, gradient orientation of the edge and HoughMap

2.4 Skin Color Detector

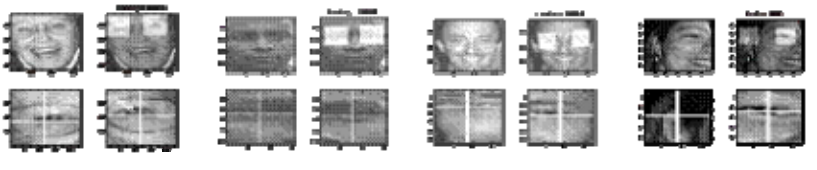
Independently of gradient's orientation information, a coarse skin detection has been implemented in the YCbCr color space [12]. Our coarse skin color filter is defined by $Cb \in [105 \quad 130]$ and $Cr \in [135 \quad 160]$. These thresholds were experimentally tuned using images with people.

This skin detector is coarse [13] and in some case no skin at all is filtered but the combination of the detectors enables us to use a simple model. At each point of the image, we compute proportion of skin-color pixels inside the 13x17 neighbor. The result map is called SkinMap.

2.5 Eyes Detector

A global transformation, referred to as Chinese Transform (CT) [14], which input is the gradient orientations of the edge pixels of the image, provides as output a cumulative arrays (so called VOTE) that gives information on the centers of the eyes, and the size of a rectangular window around each eye. True eyes location correspond fairly well to the pixels with maximum number of votes. This maximum constitutes a first eyes feature, and its use is explained in the next section.

The latter technique provides a robust but not very precise localization of the eyes. A refinement tool is required to increase detection precision according to the vertical axis. The position of the eye center corresponds to some local minimum of the vertical profile. The profile is as usual the mean value of pixel values calculated for each line in the sub-window. Several local minimum are sometimes detected and are due to the eyebrows and/or the shade under the eye. As the disk formed by the pupil is darker than its environment (even for dark peoples) it corresponds to the profile global minimum.



(a) D=0.019 (b) D=0.017 (c) D=0.025 (d) D=0.22

Fig. 3. Dissimilarity values for good (a,b,c) and false (d) detections

A validation step simply compares images of both eyes. The dissimilarity (after vertical symmetry) is calculated by normalizing grayscale and size of the two sub-windows to be compared. Then we apply a vertical symmetry (left/right flip) to the second image. The dissimilarity is the Euclidean distance between these 2 normalized and flipped windows (see fig. 3). This dissimilarity constitute the last of five face features, yielding the DissimilMap as we will see in section 3.

3 Neural Combination

These five sources of information are combined for face localization purpose. In a first stage the first three (appearance-based model, ellipse detector and skin color detector) are linearly combined to eliminate most of non-face windows. A second stage fusions the five detectors using a neural network in order to localize the face.

We have implemented three detectors for a color image, which result in three maps : DiaboloMap, HoughMap, and SkinMap. Each detector map is linearly adjusted onto $[-1 \quad 1]$. We will denote these normalized maps as D, H, and S. Using the three detectors, a pixel (i,j) in the original image is then featured by $I_{i,j} = [H_{i,j} \quad D_{i,j} \quad S_{i,j}]$. Examples extracted from 100 images of the diabolo's cross-validation set are used to learn linear combination parameters using a gradient descent stopped by cross validation :

$$FusionMap_{i,j} = a \cdot H_{i,j} - b \cdot D_{i,j} + c \cdot S_{i,j}$$

with a=0. 2280, b=0.2620 , c=0. 1229.

We can notice that the weight of the skin detector is smaller than Hough and diabolo detectors ones. This linear combination is used as a pre-filter for the eye features calculation. Each maxima of the FusionMap correspond to a sub-image in the original image. Face candidates are the windows close enough to three higher maxima (an example is given in fig. 4). We call them "face candidate windows". A window is "close enough" to a maxima when it overlap the corresponding sub-image by more than 60%.

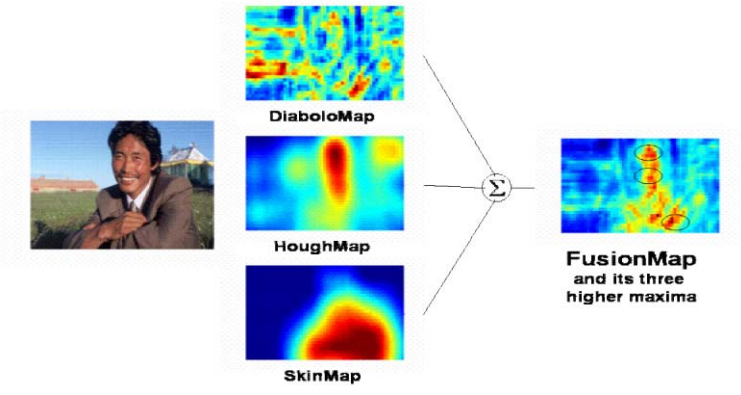


Fig. 4. Overview of the pre-filtering stage

The two eye features (dissimilarity and maximum number of votes in the eye VOTE array) are calculated over these « face candidate windows » resulting in two arrays, DissimMap and MaxVoteMap, linearly adjusted onto $[-1 \quad 1]$ (see fig. 5).

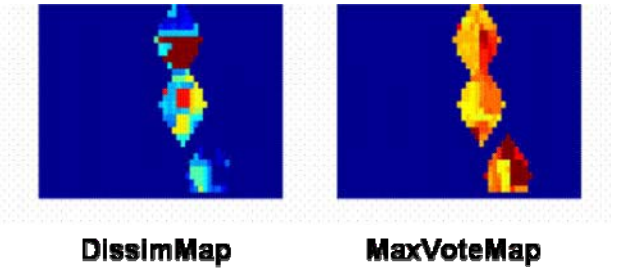


Fig. 5. Eye features calculated on face candidates resulting from the pre-filter

On our test database of 1352 images, the linear combination is calculated over 3 001 558 sub-windows, and only 271 896 sub-images are submitted to the Chinese Transform. So, the first stage eliminate **91%** of all possible windows.

This process is implemented over 155 images of the training database described earlier providing five maps used to train, with a gradient descent stopped by cross validation , a multilayer perceptron (MLP). Its architecture consists of 5 inputs, 4 hidden neurons (with a sigmoid activation function) and one output neuron (with a linear activation function).

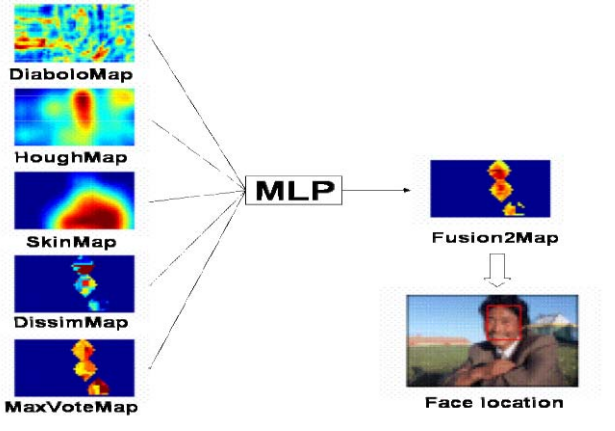


Fig. 6. Face location correspond to Fusion2Map maximum

The MLP is trained to output +1 if the input image is a face, and -1 if it's a non face.

MLP response over the complete image is called Fusion2Map, and the maximum of this map defines face location in the original image as shown in fig. 6.

4 Face Localization Performances Evaluation

Combining multiple sources greatly improves localization performances [15]. To determine face location, the input image is scanned (first by the pre-filter then by the MLP) and the location of the face is defined as the maximum of the resulting map Fusion2Map. We use the ECU face database [16] which is composed of color images (set 1), and two corresponding ground truth : one for the faces (set 2) and one for the skin (set 3). This database includes faces with various poses and skin tones.

Our test set is made of 1352 images (non overlapping with the training and cross-validation corpus) containing only one face, to evaluate localization rate. For each image, face's size is supposed to be known, which enables us to apply a 13x17 window sliding strategy, this knowledge is equivalent to knowing the distance between the camera and the person to be localized. Each image of the test set is first resized so that the face (defined by the ground truth) reaches the size 13x17. This size was chosen to respects faces aspect ratio, and is a good compromise between face's feature visibility (by human vision) and computational efforts.

A face is considered as correctly localized if the detection bounding box covers at least 60% of the area of the bounding box defined by the ground truth. To evaluate improvement brought by the combination of the 5 detectors, the proportion of pixels of the ground truth inside the detection (which is the location of the maximum for HoughMap and of the minimum for DiaboloMap) is calculated for each test image. Using the DiaboloMap minimum position alone, 656 faces are correctly located (proportion of good pixels greater than 0.6) in our test set (48.5%). Using the maximum of HoughMap alone to define the location of the face, 903 faces are correctly detected (67%).

After combination, 1197 faces (over the 1352 test images) are correctly detected which increases the detection rate to **88.5%.** The combination of the five detectors decreases the error rate of more than 50%.

5 Conclusion and Prospects

This communication aimed to present a significant contribution to the face localization task. We have presented five different detectors: skin color, auto-associative multi-layer perceptron, ellipse Hough Transform, and two eye features. A linear combination of the first three feature eliminates most of face candidate, and activates the eye detector. The five detectors are combined using a MLP and an awesome improvement of localization rate.

Several improvements are in progress: more sophisticated skin color models like ellipsoidal thresholding, Gaussian density functions [17] or mixture of Gaussians [18]. Note that the Chinese Transform is also useful in other contexts like detection of axial symmetries or the localization of patterns [14]. Our experimental results show that to solve the problem arising from tilted or slanted heads, a 3D pose detection of the head is necessary; this detection should be done without the localization of the eyes, with an holistic algorithm. Our team already test such an approach for face pose estimation which gives promising results.

References

1. Yang, M.-H, Kriegman, D., Ahuja, N.: Detecting Faces in Images: A Survey, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 1, (2002), 34-58
2. Verma, R.C., Schmid, C., Mikolajczyk, K.:Face Detection and Tracking in a Video by Propagating Detection Probabilities, IEEE Trans. PAMI, vol.25, no 10, Oct. 2003
3. Bileschi, S.M., Heisele, B.: Advances in Component Based Face Detection, IEEE International Workshop on Analysis and Modeling of Face and Gestures (2003)
4. Garcia, C., Delakis, M.: Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection IEEE Trans. PAMI, vol. 26, no. 11, Nov. 2004
5. Froba, B., Zink, W.: On the combination of different template strategies for fast face detection, Multiple Classifier Systems, (2001) 418-428
6. Rahman, A.F., Fairhust, M.C.: Multiple classifier decision combination strategies for character recognition: a review, International Journal on Document Analysis and Recognition, (5) (2003) 166-194
7. Czyz, J., Kittler, J., Vandendorpe, L.: Combining face verification experts, International Conference on Pattern Recognition, vol 2, (2002) 28-31
8. Belaroussi, R., Prevost, L., Milgram, M.: Combining model-based classifiers for face localization, to appear in Proc. of IAPR Conf. on Machine Vision & Application (2005)
9. Féraud, R., Bernier, O., Viallet, J., Collobert, M.: A Fast and Accurate Face Detector Based on Neural Networks, IEEE Trans.PAMI, vol. 23, no. 1, (2002) 42-53
10. Schwenk, H., Milgram, M.: Transformation invariant auto-association with application to handwritten character recognition, Neural Information Processing Systems 7, (1995) 991-998
11. DeMers, D. and Cottrell, G.: Non-linear dimensionality reduction. Neural Information Processing Systems 5, (1993) 580–587
12. Chai, D., Nang, K.N.: Locating facial region of a head-and-shoulders color image. International conference on Automatic Face and Gesture Recognition, (1998) 124-129
13. Hu, M., Worrall, S., Sadka, A.H., Kondoz, A.M.: Automatic scalable face model design for 2D model-based video coding, Signal Processing: Image Communication, vol. 19, (2004) 421-436
14. Milgram, M., Belaroussi, R., Prevost., L.: Multi-stage combination of geometric and colorimetric detectors for eyes localization. to appear in Proc. of ICIAP 2005
15. Prevost, L., Milgram, M.: Automatic Allograph Selection and Multiple Expert Classification for Totally Unconstrained Handwritten Character Recognition, International Conference on Pattern Recognition, vol 1, (1998) 381-383
16. Phung, S. L., Bouzerdoum, A., Chai, D.: Skin segmentation using color pixel classification: Analysis and comparison, IEEE Trans. PAMI, vol. 27, no. 1, (2005) 148-154
17. Yang, M.-H, Ahuja, N.: Detecting human faces in color images, Proceedings Of IEEE International Conference on Image Processing, vol. 1, (1998) 127-130
18. McKenna, S.J., Gong, S., Raja, Y.: Modeling facial color and identity with gaussian mixtures, Pattern Recognition, vol 31, no.12, (1998) 1883-1892

3D Face Matching Using the Surface Interpenetration Measure

Olga R.P. Bellon, Luciano Silva, and Chauã C. Queirolo*

Universidade Federal do Paraná, Departamento de Informática,
IMAGO Research Group, CP 19081, 81531-980, Curitiba - PR, Brasil
{olga, luciano, ccqs03}@inf.ufpr.br
http://www.inf.ufpr.br/imago

Abstract. 3D face recognition has gained growing attention in the last years, mainly because both the limitations of 2D images and the advances in 3D imaging sensors. This paper proposes a novel approach to perform 3D face matching by using a new metric, called the Surface Interpenetration Measure (SIM). The experimental results include a comparison with a state-of-art work presented in the literature and show that the SIM is very discriminatory as confronted with other metrics. The experiments were performed using two different databases and the obtained results were quite similar, showing the robustness of our approach.

1 Introduction

Nowadays, the researches in 2D face recognition have reached a significant degree of success but it is well known that some drawbacks still remain [1]. A number of computer vision experts believe that these limitations could be solved by using 3D information. Therefore, 3D face recognition has gained growing attention in the last years. The advances in 3D imaging technology (*e.g.* Intl. Conf. on 3-D Digital Imaging and Modeling) also have played an important role in this scenario.

Some important but few, disperse contributions regarding 3D face processing were made in early 90's [2,3,4,5]. In the last years, the contributions related to this subject have increased in number and more consistent experimental results have been presented, mostly based on private databases. For a survey of works presented in the literature about 3D face processing the reader should refers to [6].

Basically, the main approaches for 3D face recognition are based in one or combinations of different techniques, including: image registration by the Iterative Closest Point (ICP) [7] algorithm; Extended Gaussian Image (EGI) [8]; and Principal Component Analysis (PCA) [9]. Recently, ICP-based techniques have been developed [10,11] to assess matching quality based on Root Mean Square Error (RMSE). Because the RMSE alone is not discriminatory enough [10] for the purpose, these works combine other information to achieve a final decision

* The authors would like to thank CNPq for financial support.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1051–1058, 2005.

about the recognition [10]. EGI-based techniques were more applied in the early 90's and present severe limitations regarding scale [3,12]. Finally, the PCA family techniques have been extensively used for 2D face recognition but even by adding 3D information they still suffer from facial expression limitations and also there is the "curse of dimensionality". Multi-modal approaches have also applied PCA techniques [13,14].

In this work we propose a new metric for the 3D face matching problem, based on the Surface Interpenetration Measure (SIM) [15,16,17,18]. The SIM has proved to be a robust measure for 3D image registration and the experiments presented in this paper shows that it is also suitable for 3D face matching. Even more, the SIM produces a better range for discrimination between faces as compared with other metrics. In order to prove that, we reproduced the metrics presented in [10] and included comparisons against the SIM. Our experiments were performed using two different databases and the obtained results validate the robustness of our approach, which may contribute substantially to this field.

This paper is organized as follows. First, we introduce our developed approach by using the Surface Interpenetration Measure (SIM) in section 2. The experimental results by comparing different measures for 3D face matching are presented in section 3, followed by a conclusion in section 4.

2 Developed Approach Using the SIM

We propose to perform 3D face matching based on our robust range image registration method which combines Genetic Algorithms (GA) and the Surface Interpenetration Measure (SIM)[15,16,17,18].

2.1 The Surface Interpenetration Measure

The SIM was developed by analyzing visual results of two aligned surfaces, each rendered in a different color, crossing over each other repeatedly in the overlapping area. The interpenetration effect results from the nature of real range data, which presents slightly rough surfaces with small local distortions caused by limitations of the acquiring system [15]. Because of this, even flat surfaces present a "roughness" in range images. With this, we can assume that independently of the surfaces' shapes the interpenetration will always occur. We also observed that two views acquired from the same object surface with the same scanner position and parameters provide two different range images.

By quantifying interpenetration, one can more precisely evaluate registration results and provide a highly robust control. To do this we developed the following measure based on the surface normal vector, computed by a local least squares planar fit, at each point. After the alignment of two images, A and B, we identify the set of interpenetrating points in A with respect to B. For each point $p \in A$ we define a neighborhood N_p to be a small $n \times n$ window centered on p. With q denoting a point in the neighborhood N_p, c the corresponding point of p in image B and $\boldsymbol{n}_c$ the local surface normal at c, we define the set of interpenetrating

points as: $C_{(A,B)} = \{p \in A \mid [(\overrightarrow{q_i - c})\cdot \boldsymbol{n}_c][(\overrightarrow{q_j - c})\cdot \boldsymbol{n}_c] < 0\}$; where $q_i, q_j \in N_p$ and $i \neq j$. This set comprises those points in A whose neighborhoods include at least one pair of points separated by the local tangent plane, computed at their correspondents in B. With this, we then define the SIM as the fraction of interpenetrating points in A: $SIM_{(A,B)} = \frac{|C_{(A,B)}|}{|A|}$.

Registrations of two views presenting good interpenetration have high SIM's values. Our experimental results show that erroneous alignments produce low SIM's values and that small differences in MSE can yield significant differences in SIM. Furthermore, alignments with high SIM present a very low interpoint distance between the two surfaces. That is, the SIM is a far more sensitive indicator of alignment quality when comparing "reasonable" alignments [16,18].

2.2 Robust GA-Based Approach for Range Image Registration

The general principle underlying GA is to maintain a population of possible solutions (individuals) encoded in the form of a chromosome (a string of genes) and to submit the population to an evolutionary procedure, until some criteria can be satisfied [19]. In this process GA operators (*e.g.* crossover, mutation and selection) and a fitness function are used to define the best individuals for each generation. At the end of the evolutionary procedure we have the best individual.

Here the goal of the registration problem is to find through a pose-space search, rather than correspondence-base search of ICP, geometric transformations that can be used to align two views precisely. To perform range image registration using GA it is necessary to define a chromosome encoded as: 6 genes, 3 parameters each of rotation and translation [17]. With this robust GA-based approach driven by the SIM we obtained precise alignments [15,16,18].

In this work we adopted the same strategy as in [16,18] to align views to allow a precise face matching. However, as shown in section 3, the SIM did not necessarily need our robust GA-based registrations to achieve discriminatory matching results , which can also be obtained from ICP-based registrations.

Fig. 1 shows examples of using face views from two different databases: Notre Dame 3D Face Database (NDU)[1] and OSU SAMPL Database (OSUD)[2]. The alignments shown in Figs. 1(c) and 1(i) were obtained using an ICP-based approach (see section 3 for details).

3 Experimental Results for 3D Face Matching

The main goal of our experiments was to validate the SIM as a novel measure for 3D face matching. For this, we use a number of different faces from two different range image databases. Each pair of views were aligned with with two different approaches (GA and ICP) to show the SIM's improvements and robustness. Also, we compare our measure against the ones proposed in [10].

[1] `www.und.edu/~cvrl`

[2] `sampl.eng.ohio-state.edu`

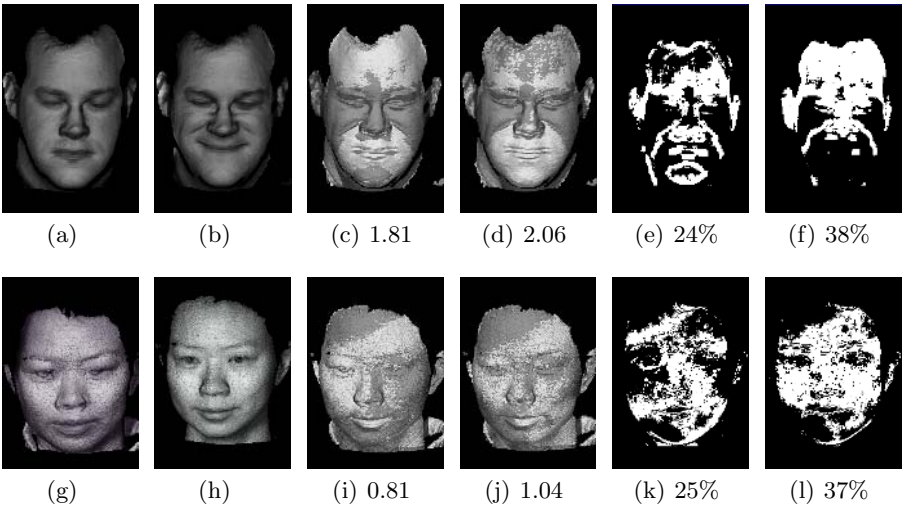


Fig. 1. Examples of SIM resuls: (a) and (b) views from OSUD; (g) and (h) views from NDU; (c and i) and (d and j) are registration results with their RMSE values using ICP-based (c and i) and GA-based approaches (d and j), respectively; (e and k) and (f and l) are the SIM values and the binary images of the interpenetrating points obtained from the alignments as show in (c and i) and (d and j), respectively

Figs. 1(d) and 1(j) were obtained using our robust GA-based approach. As can be seen in Figs. 1(c)-1(d) and Figs. 1(i)-1(j) both aligned views show correct registrations having similar RMSE values. However, by analyzing the SIM values and the respective binary images obtained by our approach we obtained more precise alignments to improve the face matching problem.

From our best knowledge, there is no large public database of 3D faces. In our experiments, we used two databases[3]: the Notre Dame 3D Face Database (NDU) and the OSU SAMPL Database (OSU). From the NDU database, we used only the portion of the image that represents the face to obtain precise measures during our experiments. This database includes 200 subjects with different views and all of them were acquired in frontal view and with neutral expression. In our experiments, we selected a set of 2 views from each one of the 5 subjects (A-E), as shown in Fig. 2. The image dimensions are 640x480, obtained by a Minolta Vivid 900 range scanner (`www.minoltausa.com/vivid`).

From the OSU database, we selected 6 subjects (A-F) as presented in Fig. 3: two of them (A and B) have 3 images with neutral facial expression acquired at 36 degree spacing as shown in Figs. 3(a)-3(c) and Figs. 3(h)-3(j), respectively; and 2 images with smiling facial expression (Figs. 3(d)-3(e) and Figs. 3(k)-3(l)); and 2 images with sad facial expression (Figs. 3(f)-3(g) and Figs. 3(m)-3(n)). All the remaining have 1 frontal image with neutral expression, as in Figs. 3(o)-3(r). The image dimensions are 200x200 obtained by a Minolta Vivid 700.

[3] The authors would like to thank the databases owners for allowing us to use them.

(a) A0 (b) A1 (c) B0 (d) B1 (e) C0

(f) C1 (g) D0 (h) D1 (i) E0 (j) E1

Fig. 2. Faces from the Notre Dame 3D Face Database

(a) A0 (b) A1 (c) A2 (d) A3 (e) A4 (f) A5

(g) A6 (h) B0 (i) B1 (j) B2 (k) B3 (l) B4

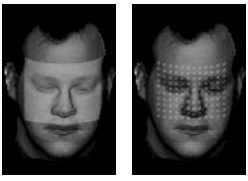
(m) B5 (n) B6 (o) C (p) D (q) E (r) F

Fig. 3. Faces from the OSU SAMPL Database

Because the range views from both databases were not registered we had to align each pair of views used in our experiments by using two different algorithms: 1) ICP-based approach driven by the RMSE measure used in the Scanalyze Software (`graphics.stanford.edu/software`); and 2) our robust GA-based approach driven by the SIM as presented in [16,18,15].

With the obtained registrations we reproduced the metrics presented in [10], which use a combination of RMSE and cross correlation of the shape index vectors [20] as a face matching measure. Also, the registrations were used to compare the SIM against different metrics as presented bellow.

First, from each aligned view pair the region R around eyes and nose (see Fig. 4(a)) was manually extracted because it is considered rigid, suffering less influence from facial expressions [21]. Then, from each region we selected a set S_{pts} of control points (around 100) equally spaced, as shown in Fig. 4(b). For both R and $Spts$ we considered only valid corresponding points within the overlap area to compute the metrics as in [10].



(a) (b)

Fig. 4. Regions of interest used to compute the matching measures as in [10]: (a) region R around eyes and nose; (b) set S_{pts} of control points

The results in applying the approach proposed in [10] on views from both databases are presented in Tables 1 and 2, respectively. The first two columns of the tables (V1 and V2) represent each aligned pair (*e.g.* view A0 aligned to A1). M1 represents the global RMSE of the alignment; M2 is cross correlation of the shape index vectors and M3 the face matching measure as proposed by [10], which is the combination of M2 and the RMSE in region R. The RMSE in the region R can be known by: $M_R = M3 - M2$.

As suggested in [22], a pre-defined threshold $\delta = 1$ should be used to classify correct and incorrect matchings. If $M3 < \delta$ than we have a correct matching, *i.e.* alignment A0-A1 from Table 1; otherwise the aligment represents an incorrect matching, *i.e.* aligment B0-E1 from Table 1. However, it is well known that threshold definition is not a trivial process. We observed that in some cases wrong matching occurs (*e.g.* alignments B0-B1 and B0-B2 from Table 2), which are faces from the same subject with neutral expression. Also, because the range between correct and incorrect matchings tends to be lower it is hard to define a good discriminant threshold.

By analysing M3 values we observed that the distance metrics used in this approach fails in detecting correct matchings for faces of a same subject but with different facial expression. For instance in the Table 2, the M3=1.58 of alignment B0-A1 (distinct subjects) should be higher than the results for the alignment A0-A3 (same subject), which have M3=1.85. In contrast, for the SIM, the difference between correct and incorrect matching values are much higher (for instance alignments C0-C1 and B0-E1 from Table 1). Also, the correct matches for two images of the same subject but with different facial expressions have an intermediate value between the correct and incorrect matches, suggesting that our matching measure can deal with changes in facial expressions.

We also performed a number of experiments to evaluate the SIM's results in terms of ICP-based and GA-based alignments. Tables 1 and 2 show the results of the SIM for both approaches: columns S1 and S1g presents the SIM's results in the region R for the ICP-based and GA-based alignments, respectively. From the SIM's equation A and B represent the views V1 and V2, respectively. The columns S2 and S2g present the global SIM values of the alignments. As can be seen, the results generated by the ICP-based approach are quite similar from those obtained by applying our GA-based approach, proving that the applied metric (S1 or M3) is more relevant than the registration technique.

Table 1. Experimental results from the Notre Dame 3D Database

V1	V2	M1	M2	M3	S1	S1g	S2	S2g
A0	A1	0.85	0.04	0.72	27.4	29.0	24.7	29.8
B0	B1	0.81	0.04	0.67	30.0	58.7	24.9	36.7
C0	C1	0.98	0.02	0.84	26.3	38.4	22.6	30.8
D0	D1	0.65	0.04	0.64	28.5	31.6	30.5	36.3
E0	E1	0.68	0.04	0.57	42.1	42.4	47.3	49.1
A0	B1	2.18	0.08	2.25	0.62	0.89	2.19	3.67
A0	C1	2.24	0.07	2.08	4.50	6.82	3.64	8.58
A0	D1	2.28	0.06	2.25	1.82	6.39	3.63	8.26
A0	E1	2.43	0.08	2.42	1.85	1.64	1.47	2.22
B0	A1	2.23	0.09	2.60	1.38	2.83	3.19	5.52
B0	C1	2.37	0.10	2.34	1.35	3.64	3.24	6.51
B0	D1	2.34	0.05	3.20	1.14	7.71	2.83	6.63
B0	E1	2.02	0.06	1.24	7.87	10.1	6.51	10.2
C0	A1	2.16	0.08	1.94	5.92	3.52	4.95	7.71
C0	B1	2.65	0.11	2.75	0.10	0.54	1.40	2.78
C0	D1	1.74	0.05	1.97	6.47	6.02	8.31	9.19
C0	E1	2.35	0.06	1.42	1.02	7.14	1.07	3.88
D0	A1	2.17	0.07	1.67	1.13	3.78	3.30	7.88
D0	B1	2.42	0.05	2.04	2.76	1.50	1.89	4.37
D0	C1	1.78	0.08	1.91	4.91	10.7	7.18	10.3
D0	E1	2.12	0.13	2.51	2.69	6.04	2.42	4.73
E0	A1	2.37	0.13	2.09	1.70	5.25	1.96	3.49
E0	B1	1.96	0.11	2.28	6.54	7.66	5.13	12.9
E0	C1	2.15	0.06	2.40	4.74	7.73	4.66	7.25
E0	D1	2.15	0.04	1.82	3.98	2.66	4.60	6.43

Table 2. Experimental results from the OSU SAMPL Database

V1	V2	M1	M2	M3	S1	S1g	S2	S2g
A0	A1	0.99	0.01	0.93	59.5	62.2	57.6	60.8
A0	A2	0.85	0.03	0.93	71.7	72.0	62.8	65.5
B0	B1	1.04	0.01	1.14	62.2	63.8	67.8	68.4
B0	B2	1.05	0.01	1.04	53.6	51.5	55.4	57.8
A0	A3	1.81	0.06	1.85	27.3	35.4	23.7	37.7
A0	A4	1.85	0.06	2.08	18.5	21.0	17.4	21.6
A0	A5	1.27	0.04	1.45	31.0	36.9	32.5	35.8
A0	A6	1.20	0.02	1.68	28.0	32.4	27.4	32.8
B0	B3	1.70	0.02	1.78	34.3	32.5	30.0	33.5
B0	B4	1.69	0.03	1.85	25.4	32.3	20.5	28.5
B0	B5	1.91	0.04	1.87	35.0	51.2	24.7	34.3
B0	B6	1.93	0.02	1.77	25.4	36.5	18.9	26.3
A0	B0	2.16	0.11	2.70	9.07	10.6	12.2	12.5
A0	B1	2.08	0.13	2.61	9.97	8.57	11.7	13.6
A0	B2	2.12	0.08	2.77	7.02	9.69	9.81	11.1
A0	C	2.21	0.10	2.31	6.33	11.0	5.53	7.59
A0	D	2.25	0.10	2.43	8.23	5.98	5.47	8.09
A0	E	2.18	0.08	2.27	4.19	7.34	4.33	6.56
A0	F	2.21	0.12	2.69	10.7	8.95	8.75	13.5
B0	A1	2.39	0.06	1.58	4.32	6.78	7.11	8.09
B0	A2	2.17	0.05	1.61	9.57	9.41	10.5	11.4
B0	C	2.65	0.11	2.88	1.69	1.71	3.32	4.61
B0	D	2.44	0.07	2.16	3.10	9.32	3.46	6.64
B0	E	2.66	0.11	2.98	1.90	2.01	4.64	5.06
B0	F	2.19	0.05	2.07	9.16	10.2	10.7	12.9

From the tables, one can see that for the RMSE values the range between a correct matching (views form the same subject) and an incorrect one (views from different subjects) is very small, although they fit the values specified in [10].

4 Conclusion

In this paper we introduce a novel and robust approach for 3D face matching by using the Surface Interpenetration Measure. We observed that the SIM is a reliable, discriminant measure that can deal better with changes in facial expressions as compared with other measures (*i.e.* M3). Although the SIM's results from our GA-based approach have presented better results, we proved that the SIM, when used as a face matching measure, works even with ICP-based alignments. As a future work we plan to use the SIM to identify face expressions and to develop a robust approach for 3D face identification problem using a larger database.

References

1. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys **35** (2003) 399–458
2. Achermann, B., Jiang, X., Bunke, H.: Face recognition using range images. In: Intl. Conf. on Virtual Systems and Multimedia. (1997) 129–136
3. Lee, J.C., Milios, E.: Matching range images of human faces. In: Intl. Conf. on Computer Vision. (1990) 722–726

4. Negamine, T., Uemura, T., Masuda, I.: 3d facial image analysis for human identification. In: Prof. of ICPR. (1992) 324–327
5. Brunelli, R., Falavigna, D.: Person identification using multiple cues. IEEE PAMI **17** (1995) 955–966
6. Bowyer, K.W., Chang, K.I., Flynn, P.J.: A survey of approaches to three-dimensional face recognition. In: Proc. of ICPR. (2004) 358–361
7. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE PAMI **14** (1992) 239–256
8. Horn, B.K.P.: Extended gaussian images. In: DARPA84. (1984) 72–89
9. Martinez, A.M., Kak, A.C.: Pca versus lda. IEEE PAMI **23** (2001) 228–223
10. Lu, X., Colbry, D., Jain, A.K.: Three-dimensional model based face recognition. In: Proc. of ICPR. (2004) 362–366
11. Cook, J., Chandran, V., Sridharan, S., Fookes, C.: Face recognition from 3d data using iterative closest point algorithm and gaussian mixture models. In: 3D Data Processing, Visualization, and Transmission. (2004) 502–509
12. Tanaka, H.T., Ikeda, M.: Curvature-based face surface recognition using spherical correlation - principal directions for curved object recognition. In: ICPR. Volume 3. (1996) 638–642
13. Chang, K.I., Bowyer, K.W., Flynn, P.J.: An evaluation of multimodal 2d+3d face biometrics. IEEE PAMI **27** (to appear in 2005) 619–624
14. Chang, K.I., Bowyer, K.W., Flynn, P.J.: Multi-modal 2d and 3d biometrics for face recognition. In: Intl. Workshop on Analysis and Modeling of Faces and Gestures. (2003) 187–194
15. Silva, L., Bellon, O.R.P., Boyer, K.L.: Robust Range Image Registration Using Genetic Algorithms and the Surface Interpenetration Measure. Volume 60 of Machine Perception and Artificial Intelligence. World Scientific Publishing (2005)
16. Silva, L., Bellon, O.R.P., Boyer, K.L.: Robust range image registration using the surface interpenetration measure and enhanced genetic algorithms. IEEE PAMI **27** (2005) 762–776
17. Silva, L., Bellon, O.R.P., Gotardo, P.F.U., Boyer, K.L.: Range image registration using enhanced genetic algorithms. In: IEEE International Conference on Image Processing. Volume 2. (2003) 711–714
18. Silva, L., Bellon, O.R.P., Boyer, K.L.: Robust multiview range image registration. In: Proc. of 17th Brazilian Symposium on Computer Graphics and Image Processing. (2003) 80–88
19. Man, K.F., Tang, K.S., Kwong, S.: Genetic algorithms: concepts and applications. IEEE Trans. on Industrial Eletronics **43** (1996) 519–534
20. Dorai, C., Jain, A.K.: COSMOS - a representation scheme for 3d free-form objects. IEEE PAMI **19** (1997) 1115–1130
21. Colbry, D., Lu, X., Jain, A.K., Stockman, G.: 3d face feature extraction for recognition. Technical Report 4-39, MSU-CSE (2004)
22. Lu, X., Jain, A.K.: Integrating range and texture information for 3d face recognition. In: IEEE Workshop on Applications of Computer Vision. (2005) 156–163

Automatic Recognition of Road Sign Passo-Carrabile

Luca Lombardi, Roberto Marmo, and Andrea Toccalini

Dipartimento di Informatica e Sistemistica, University of Pavia, 27100 Pavia, Italy
{luca.lombardi,roberto.marmo}@unipv.it
http://vision.unipv.it

Abstract. This paper describes a method to detect and identify the typical italian road sign passo-carrabile. The system first determines the region of interest within the image, using color segmentation, then the signal of no waiting is identified using shape and color information, and finally the text label passo-carrabile is recognised with a state diagram and a set of specific tests on the shape of the words. The obtained results show the feasibility of the system.

1 Introduction

Discriminations between road signs can be performed mainly by its colors, which are chosen to identify the sign easily and differentiate it from the background. According to this, the first analysis consists in color segmentation. One problem is the high sensibility of colors to weather changes, so this first elaboration is applied keeping a high tolerance. The italian road sign Passo-carrabile can have different dimensions, but the standard model (Fig. 1) is 24 cm wide and 43 cm long, with a no waiting signal of 10 cm radius; this sign is painted only with red, blue, white and black colors [9]. This specific sign gives the order to keep clear a transit zone, it forbids parking in a lateral area of the street located into town, so other objects with similar colors can make more difficult the recognition.

It is important to examine also the text label passo-carrabile as an additional test and to differentiate this specific sign among all other prohibition signs.

The algorithm created to find the road sign is defined on three levels. First, the program performs a rapid analysis of the whole image, and selects the area that will be analyzed more accurately, according to the distribution of the sign colors, next levels work on the selected region; in case of negative response the control turns back to the first level, and the research continues on the image. If the previous constrains are satisfied, we use the Hough Transform to find red and blue circles, which correspond to disks of the prohibition signal: if circles are detected and they have a common center, we use specific measurements to estimate the real dimensions of the sign. If obtained results are admittable, the program continues with more detailed controls, so we apply a pattern matching of the prohibition signal with a model, and the analysis of the text label 'passo carrabile' with a set of specific tests.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1059–1067, 2005.

Fig. 1. The road sign passo-carrabile

2 Color Segmentation

Color segmentation is the most critical point of the algorithm: it is the first operation executed, so it is very important to keep high tolerance, to avoid loosing useful informations.

The color space RGB because is the typical model used by digital cameras, so we do not need any transformation. One problem working with the RGB representation is the high sensibility to lighting conditions, so we study relations between color components [1,4,5,10]. Converting the RGB space to HSV (Hue, Saturation, Value) allows to gain a better control on chromatic variations of each hue (it is a separated component), but the computational cost is prohibitive, because the space transformation is nonlinear. HSV space is very tipic [3,6,7,8,11], but some studies shows that Hue component changes considerably according to distance, brightness and sign age. Thus we decided to use the RGB model to reduce execution time, also because we analyze only two colors (red and blue) and these are represented by two separately channels in RGB space; then white color can be easily obtained putting together the three components.

Now we extract a region of the image with enough quantity of these colors. Three color image histograms (Fig. 2) are analyzed to find a point with a sufficient number of colored pixels, of the 3 interested colors, on the same row and also on the same column [12].

3 Analysis of the Region of Interest

The algorithm for circles detection works on binary images, the blue and red masks of the inspected image area. We have refined the colors segmentation, adjusted on the mean brightness of the analyzed region. A gaussian filter (mean value among adjacent pixels) removes noise, also deleting a lot of isolated pixels which would increase noise for Hough method.

Now the edges of the two binary images are extracted: for this operation we use the isotropic operator, which consists in the convolution of the image with 3x3 masks.

Hough transform then is performed to locate red and blue circles. We do not know a priori the radius of these circles, so we search in a three-dimensional parameter space to find the three values (x_c, y_c, r) corresponding to circumference. We apply the Hough method either on red and blue circles; it allows to

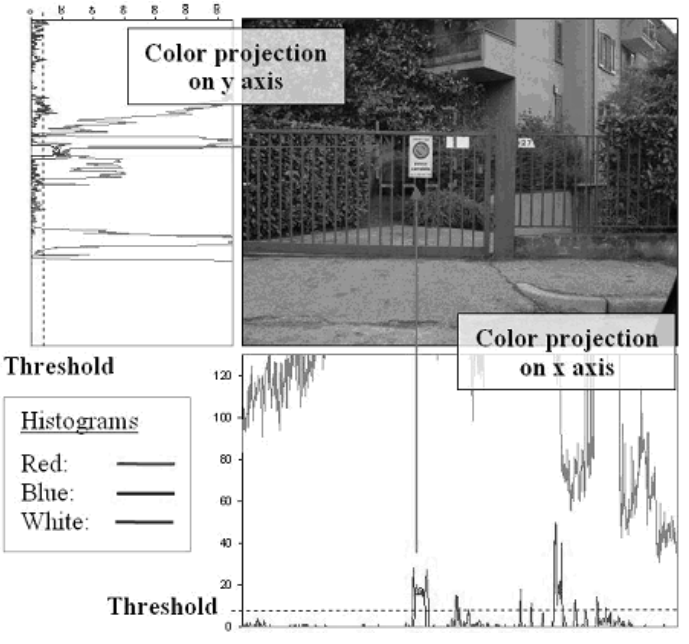


Fig. 2. Extraction of the interesting image area using histograms of colors red, blue and white

detect the center of the signal with good assurance and approximate sign dimensions. We needed to perform some optimizations to get this algorithm as lighter as possible. Now, using these results we want to find exactly the position and dimensions of the signal painted on the road sign. The idea consists in performing 8 measurements on the signal, obtaining a good strength even if we have occlusions, and also verifying the results accuracy in most common cases.

Starting from the center found with Hough, we execute 4 scannings (right, left, up and down directions) and we store positions of color changes, from blue to red and then to white. So we can identify the dimensions of the three different regions within the signal, from these values we can easily detect the radius of the blue and the red rings and obtain the radius of the signal (Fig. 3).

At this point we know exactly the position and the dimension of the signal, so we can apply a pattern matching [6, 10] with a sample image [11] (Fig. 3).

We modify the image which contains the sign to verify, to improve the results of this technique: a coloration is applied before pattern matching, every colored pixel is altered to its pure tonality (for example: R=255, G=0, B=0 for every red pixel). By this way we have good results even if the road sign is dirty or affected by bad lighting.

Finally, we apply a further control on the no-waiting signal, to identify and exclude the no-stopping signs: these are very similar signs, the main difference is a double red strip instead of a single one (Fig. 4). These signs are able to pass the pattern maching test with positive result. The approach consists in counting

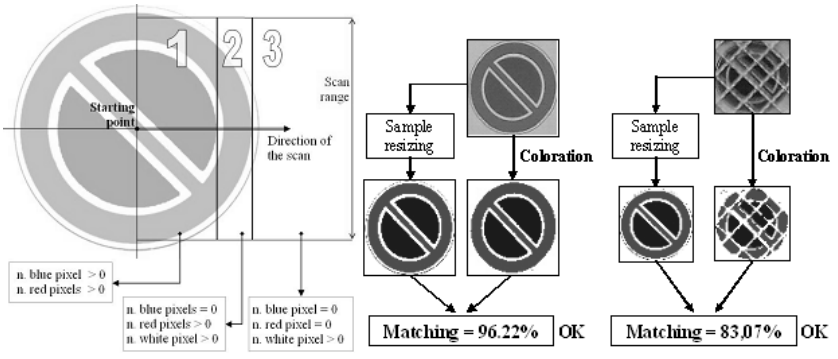


Fig. 3. Extraction of sign dimensions: on the left horizontal scan to right direction, on the right examples of pattern matching

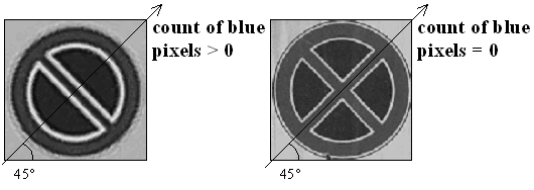


Fig. 4. No-waiting (left) and no-stopping (right) signs

the number of blue pixels on the diagonal placed at +45. When the signal has a double red strip we obtain that this number is equal to zero, in the other case the number of blue pixels always be major then zero. This procedure always given the right result.

4 Text Extraction

The text label passo-carrabile is located in a specific region, under the signal of no-waiting on the bottom of the sign. It is possible to identify characters using an O.C.R. method, but in this case we realized that it was sufficient to apply some controls on writing characteristics, there is no need to recognize each letter: thus, we can semplify and speed up the execution. Moreover, if the road sign is placed far enough from the acquisition source, the writing is displayed with very low resolution and so it is impossible to recognize exactly each character.

4.1 Pre-elaboration

We need to correct brightness and contrast of this zone, to get a better image with a clear and well-defined writing (Fig. 5). Then the image is binaryzed according to a threshold value, to simplyfy next operations like characters extraction and testing. We use dynamic parameters to perform these adjustments to the image: they are related to the mean brightness of the region analyzed.

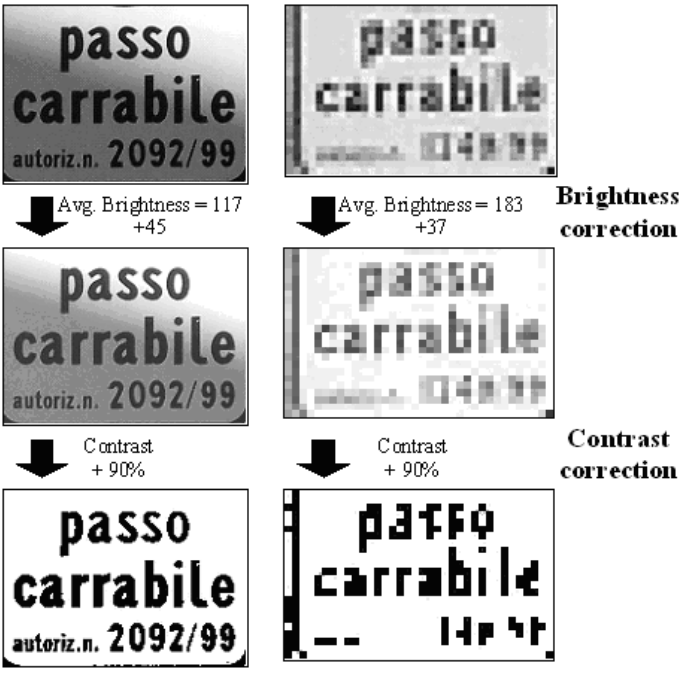


Fig. 5. Brightness and contrast adjustment

4.2 Words Extraction

To identify the exact dimensions and positions of the two words, passo and carrabile, we execute a vertical scansion of the binary image, analyzing the black projection on the y axis [2]. Working with a finite state machine we can detect both position and height of the two words (which are always one above the other), as we examine the percentage of black pixels located on each row. Sometime the sign is placed beyond a gate of the house and we obtain a noise caused by gate. We can find out this constant error performing the previous scanning, and we achieve good results even if the sign is corrupt in this way.

We can identify the error ϵ using informations from the first derivative of y projection, by the incremental ratio: $y'(i) = y(i+1) - y(i)$ where y vector contains, for each element i, the count of black pixels situated on the image row i; vector y' represents its derivative, i.e. the variation of of black pixels between two adjacent rows. Examining the vector y', we can note positive peaks corresponding to upper boundaries of words, and negative peaks on words ends: our technique consists in detecting these peaks (so locating where the words are) and extract e as the minimum value $y(i)$ among all rows i, outside these words, where $y'(i) = 0$. Briefly, we calculate the fixed error as the minimum number of black pixels located on white rows; we also assume that this noise vertically involves all the image (Fig. 6). So, we obtain two images which fits exactly the two words we have to examine.

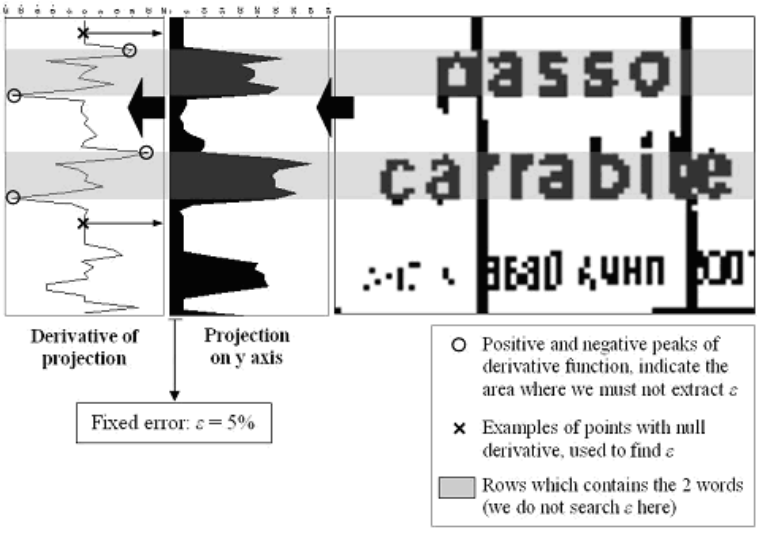


Fig. 6. Extraction of a fixed noise, using the derivative of projection on y axis

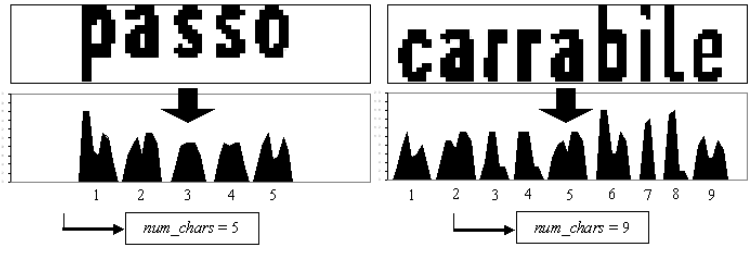


Fig. 7. Projection of black pixels on x axis

4.3 Characters Extraction

Using horizontal scans on the two words separately we can extract the number of characters, their dimensions and width analyzing the count of black pixels on each column of the image (Fig. 7). We use again a state diagram, where the transition from states 'character' and 'space' is defined by the percentage of black pixels on each column. We refine this technique, since we often have low resolution or noise on the text: for example it is useful to verify the width of each character in comparison with the median width of detected characters, realizing if characters are too large or too tight.

4.4 Final Tests

We perform some specific tests, each one allows only a little tolerance of dissimilarity from the model of a standard sign. The identification of the label is

verified succesfully if sufficient number of tests have positive results (not necessarily everyone), so we can also remove a high number of false positives (thanks to the low tolerance of each test). Briefly, this method allows to recognize the label passo-carrabile if it has enough common features with a sample image and using this tests:

1. analogy of heights of the 2 words;
2. word 'carrabile' larger then word 'passo';
3. width of the 2 words proportionate;
4. all characters of 'passo' with adequate width;
5. all characters of 'carrabile' with adequate width;
6. count of characters from 'passo' equal to 5;
7. count of characters from 'carrabile' equal to 9.

Each test has a different weight, relative to its importance for the recognition; if we still have an uncertain result after these 7 tests, we apply other 6 verifications:

8. width of word 'passo' proportional to width of the sign;
9. height of word 'passo' proportional to height of the sign;
10. matching between the 2 characters 's' from the word 'passo';
11. matching between the 2 characters 'r' from the word 'carrabile';
12. matching between the word 'passo' and a sample image;
13. matching between the word 'carrabile' and a sample image;

Table 1. Confusion matrix: c_{ij} coefficient at row i and column j represents the percentage of sign of class i identified as sign of class J

predicted classification	real classification	
	positive	negative
positive	204	6
negative	0	210

Fig. 8. Examples of right results

5 Results

Results produced by the developed system are satisfying, as we recognize correctly the signs of passo-carrabile under different lightness conditions and also with high noise. We have tested the program on a set composed of 420 images, with a resolution of 1024x768, on a computer with a Intel Pentium 4 processor at 3GHz and 512MB of RAM DDR memory at 400MHz. Concerning the computational times, the mean execution time is 74 ms and the frame per second is 13.5.

The confusion matrix (Tab. 1) shows that system accuracy is equal to 98.6%. As we can see, the program works very good with false positives: thanks to the low tolerance of last tests on the label passo-carrabile we do not have any false positive. The system is very strong against noises and occlusions of the signal, as we recognize the road sign even in very prohibitive circumstances (Fig. 8). The few failures are caused by too heavy noise on the writing or excessive faded colors.

The testing has been performed acquiring images at different distances from the sign, from 1 meter to more then 10 meters far. Sometimes the road sign is placed far from the acquisition source, so we need high image resolution to text recognition. Using a resolution of 1024x768 we experienced good results and text can be correctly detected until a distance of 11.5 meters, characters are 2 pixels wide and 3 pixels high and the sign radius measures only 11 pixels.

References

1. Escalera, A., Moreno, Salichs, Armigol.: Road Traffic Sign Detection and Classification. IEEE Transactions on Industrial Electronics 44 (1997) 848-859
2. Miura, J., Kanda T., Shirail Y.: An Active Vision System for Real-Time Traffic Sign Recognition. Proc. of IEEE Conference on Intelligent Transportation System (2000) 52-57
3. Escalera, A., Armingol, J., Mata, M.: Traffic sign recognition and analysis for intelligent vehicles. Image and Vision Computing 11 (2003) 247-258
4. Zadeh, M., Kasvand, T., Suen, C. Y.: Localization and Recognition of Traffic Signs for Automated Vehicle Control Systems. Conference on Intelligent Transportation Systems, Pittsburgh, PA, (1998) 272-282
5. Bnallal, M., Meunier, J.: Real-time color segmentation of road signs. IEEE Canadian Conference on Electrical and Computer Engineering, Montral, Qubec, Canada, (2003) 1823-1826
6. Vitabile, S., Pilato, G., Pollaccia, G., Corbello, F.: Road Signs Recognition Using a Dynamic Pixel Aggregation Technique in the HSV Color Space. Proceedings of 11 International Conference on Image Analysis and Processing, Palermo, Italy, (2001) 572-577
7. Gavrila, D. M.: Traffic Sign Recognition Revisited. Proceedings of the 21st DAGM Symposium fr Mustererkennung, Springer Verlag, Bonn, Germany, (1999) 86-93
8. Liu, H. X., Ran, B.: Vision-Based Stop Sign Detection and Recognition System for Intelligent Vehicles. Transportation Research Record 1748 (2001) 161-166
9. Italian Highway Code, http://digilander.libero.it/ordinanze/cds/codice.htm

10. Sekanina, L., Torresen, J.: Detection of Norwegian Speed Limit Signs. Proceedings of 16th European Simulation Multiconference, Darmstadt, Germany, (2002) 337-340
11. Vitabile, S., Gentile, A., Dammone, G., Sorbello, F.: MLP Neural Network Implementation on a SIMD Architecture. Lecture Notes in Computer Science 2486, Springer-Verlag, (2002) 99-108
12. Piccioli, G., DeMicheli, E., Campani, M.: A Robust Method for Road Sign Detection and Recognition. Proceedings of Third European Conference on Computer Vision, (1994) 495-500

Document Image De-warping Based on Detection of Distorted Text Lines

Lothar Mischke[1] and Wolfram Luther[2]

[1] Eduard Spranger Vocational School,
D-59067 Hamm, Vorheider Weg 8, Germany
lothar.mischke@esb-hamm.de
[2] Institute of Computer Science and Interactive Systems,
University of Duisburg–Essen,D-47048 Duisburg, Lotharstr. 65, Germany
luther@informatik.uni-duisburg.de

Abstract. Image warping caused by scanning, photocopying or photographing a document is a common problem in the field of document processing and understanding. Distortion within the text documents impairs OCRability and thus strongly decreases the usability of the results. This is one of the major obstacles for automating the process of digitizing printed documents.

In this paper we present a novel algorithm which is able to correct document image warping based on the detection of distorted text lines. The proposed solution is used in a recent project of digitizing old, poor quality manuscripts. The algorithm is compared to other published approaches. Experiments with various document samples and the resulting improvements of the text recognition rate achieved by a commercial OCR engine are also presented.

1 Introduction

In the context of a digitization project initiated in 2001 [2], for the German reception of the philosopher Nietzsche at the University of Duisburg–Essen, text documents composed between 1865 and 1945 have been digitally converted. The conversion process has been shown to be both complex and time-consuming. Many of the documents have been printed using one of the German fraktur typefaces which are not recognized very well by today's OCR programs and therefore have to be corrected manually. Due to the poor quality of some of the literary sources, which consist predominantly of photocopies of the original documents, the process of digitizing requires further human interaction. Shade, skew and warping of the document images markedly decrease OCR recognition accuracy.

To optimize the process of selecting and capturing the text documents [4] in the context of the funded project *eCampus Duisburg* a sub-project entitled "Distributed mobile selection and evaluation of documents in libraries and archives" was initiated in 2002 [3]. Mobile users digitize documents which are transmitted to a document server via WLAN. A team of experts accesses the documents via

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1068–1075, 2005.

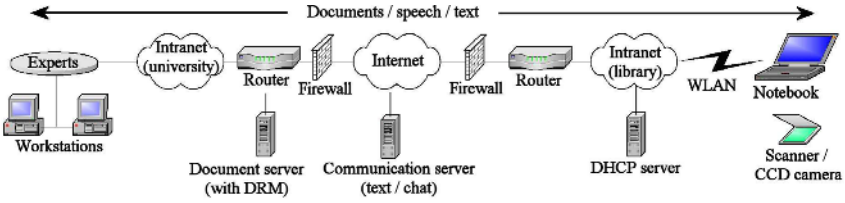


Fig. 1. Distributed document selection and evaluation (network view)

Internet, coordinates the selection, and decides about further proceedings. The network view of this process is shown in Fig. 1.

In this paper we address the preprocessing for successful OCR recognition in the case of warped document images. A brief discussion of recently published approaches to image de-warping can be found in section 2. In section 3 we present a new algorithm which is able to correct warping based on the detection of distorted text lines. Our method is capable of removing shade, correcting global skew and de-warping distorted text blocks within the document. Experiments with various document samples and the resulting improvements of the recognition rate received by a commercial OCR engine are presented in section 4.

2 Recent Approaches to Document De-warping

In the last few years several restoration methods for warped document images have been reported. These algorithms can roughly be divided into two categories:

- algorithms which make use of information derived from the source of the document distortion, and
- algorithms that simply detect the (horizontal) distortion by means of an analysis of the given document image.

The advantage of the former group of algorithms lies in the fact that knowledge of the kind of degradation within the document image can be used to model the geometrical type of distortion very well. In [5] Cao et. al. propose a cylindrical model for the restoration of camera-captured documents. Their correction algorithm is restricted to cases where the generatrix of the cylindrically assumed book surface parallels the image plane. Zhang et. al. suggest in [11] an algorithm which is capable of de-warping documents scanned from thick, bound volumes using an image scanner provided that the book spine lies parallel to the scanning light. Thus, today's model-based algorithms have the drawback that their usability is very limited as they need a lot of a priori knowledge. Currently there is no generic model which can be used for identifying and rectifying automatically all common types of warped document images.

Algorithms of the second kind do not require explicit knowledge regarding the source of distortion. Wu and Agam [9] developed a method which detects and traces curved text lines within single document images by minimizing the *local* cumulative projection within a given range of angles. The algorithm starts at the

left-hand border of a given region (which is assumed to be approximately vertical and must be provided manually) and gradually traces the curved lines. These lines are used to reconstruct a target mesh which can be used for de-warping. In [9] they apply their algorithm to perspectively degraded documents which have been captured using a digital camera. They do not impose any constraints regarding the angle between document and image plane of the camera. On the other hand results presented using this approach still show perspective distortion of the characters within the warped regions of the documents.

In [10] Zhang and Tan propose an algorithm which detects distortion by distinguishing between the light and the shaded area of a scanned gray-level document image from a bound volume. As the warped part of the image resides in the shaded area the alignment of the connected components which form a curved text line is approximated by two quadratic functions. Thus the line parts within the non-shaded area can be bound by straight reference lines. The relative position of the connected components within the shaded area with the two curves is used to move the components vertically to the corresponding straight reference lines within the non-shaded area. The orientation of the moved components is then corrected using the average angle of the tangent of the two reference curves.

Both the model-based and the analyzing type of algorithms have been shown to be suitable for increasing OCR accuracy. Although results presented indicate that the remaining distortion in the rectified documents is higher with algorithms of the second type we regard that their broader applicability is a distinct advantage. However, since within our project the original document sources are often not available and the process of digitizing and processing the documents is distributed between multiple individuals and locations (see Fig. 1), we chose to develop a method which does not rely either on explicit knowledge about the digitizing technique used (photographing, scanning, etc.) or on assumptions regarding the quality of the document image.

3 New De-warping Algorithm

The presented algorithm consists of three preprocessing steps and the document rectification step itself which are discussed in the following subsections. Fig. 2 briefly illustrates these four building blocks.

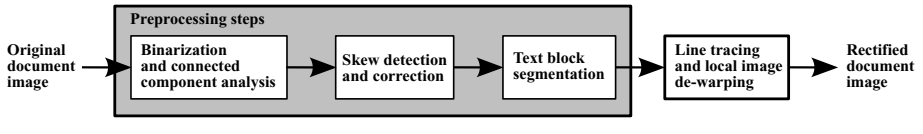


Fig. 2. Block diagram of the algorithm

3.1 Preprocessing Steps

Since most of the algorithms used to process the document image were designed for binary images we have to separate the background from the data. Warped

images often result from flatbed-scanning bound volumes. This leads additionally to a shaded region around the book spine. In this case global thresholding is ineffective. Savakis suggests in [8] an adaptive thresholding method where a document is binarized by moving an M-size window over the image, clustering all inner pixels into a foreground and a background cluster using a global threshold, and determining a local threshold for the window's central pixel by averaging these two clusters. Unfortunately this technique fails in the shaded regions of the document where the background level already exceeds the global threshold.

Therefore we divide the document image into a set of rectangular regions R_{ij}. For each R_{ij} we calculate a threshold $t_{c_{ij}}$ using Otsu's method [7] which we assign to the centroid c_{ij} of the region. Then a *localized* threshold t_{xy} for each pixel (x, y) obtained by a linear interpolation of the thresholds of the surrounding region centroids can be used for adaptive foreground/background clustering. This approach is justified by the fact that the background level decreases continuously with increasing distance to the spine area.

After binarizing the document the *connected components* must be determined. All eight-connected black pixels are grouped together in rectangular bounding boxes each of which identifies a single connected component. We use a modified version of the classical approach of Fletcher and Kastouri which is described in detail in [6] in order to separate text from graphics and to remove noise. Larger components are removed by an area/ratio filter. The remaining components are kept for subsequent use.

Skew is another standard problem in document processing and has a detrimental effect on the analysis of documents. A good survey on this topic can be found in [1]. Classical approaches make use of the Hough Transform or projection profiles in order to detect straight lines perpendicular to an accumulator line at different projection angles. As the text lines in a warped image are (partly) distorted the quality of these methods degrades. We use a skew correction method basing on projections of the centroids of the detected connected components provided that within the warped image there are still parts of the text lines which are approximately straight. Typical warping caused by scanning or photographing satisfies this constraint (see Fig. 5). This allows us in contrast to Zhang's and Tan's method [10] or the approach of Wu and Agam [9] to process document images which are *both* skewed and warped without further user interaction. The skew correction will not solve the distortion problem itself. But it aligns the nearly straight parts of the text lines parallel to the horizontal document bounds which is a precondition for the line tracing algorithm described in subsection 3.2.

After having separated text from graphics the *document structure* must be analyzed. The document is divided into text blocks which can be processed individually. For this task we use a pyramid algorithm which constructs a pyramid of less detailed images by successive reduction of the dimensions by half at each step. This leads to a natural clustering of the eight-connected components. The resulting connected components represent text blocks or isolated words which can be grouped to form text blocks.

3.2 Line Tracing and De-warping

In order to detect warped lines we identify the location of possible text lines and trace their run to the boundaries of the surrounding text block. The granularity parameter $g = 0.2$ and the angular range parameter $\theta = 15$ degrees which are used in the following steps are resolution and font independent and have been derived from experiments with multiple training documents containing varying types of distortion. Due to the preliminary skew correction we can assume that the non-distorted (straight) line parts approximately parallel the text block's horizontal borders. Thus we calculate a modified projection profile of the centroids of the connected components' bounding boxes. Let H_{avg} denote the average height of components in the text block. With text block height h we scale down the size of the accumulator to $h/(g \cdot H_{\text{avg}})$. For each bounding box a weighted vote v is calculated and added to the appropriate accumulator slot according to $v = w_j \cdot \max(0, 1 - |(h_j - H_{\text{avg}})/H_{\text{avg}}|)$, where w_j and h_j denote width and height of the bounding box, respectively. This is motivated by the observation that the line run can best be detected using average-sized bounding boxes which mostly contain the characters that lie *between* the text baselines. Notice that no further information (like manually provided vertical border lines or the location of shaded regions) is required. Fig. 3 illustrates this approach.

After processing each component the local maxima of the accumulator indicate the vertical position of the straight text line parts. As the warped parts typically lie next to the document borders we choose the horizontal middle of the text block to start the line trace. Hence for each line let (x_s, y_s) denote the starting point which we call *seed*. Then the line is built by creating an empty bounding box of height H_{avg} around (x_s, y_s) and extending it to the left and right by gradually adding all adjacent components provided that their height lies between $(1-g) \cdot H_{\text{avg}}$ and $(1+g) \cdot H_{\text{avg}}$ and y_s lies within the vertical extension of the component's bounding box. Let $(x_i, y_i), 1 \leq i \leq n$, denote the n recognized character box centroids sorted from left to right. The core zone of the text line is then set to a rectangle of width $w = r_n - l_1$ and height H_{avg}. l_1 and r_n denote the left and right bound of the leftmost and rightmost connected component, respectively. The rectangle is centered around $(x_{s'}, y_{s'})$ where $x_{s'} = (l_1 + r_n)/2$ and $y_{s'} = \text{median}_{1 \leq i \leq n}(y_i)$. The detected lines are now extended to the left and to the right by further adding adjacent components partly lying within an angular range of $\pm\theta$ degrees around the current text line orientation.

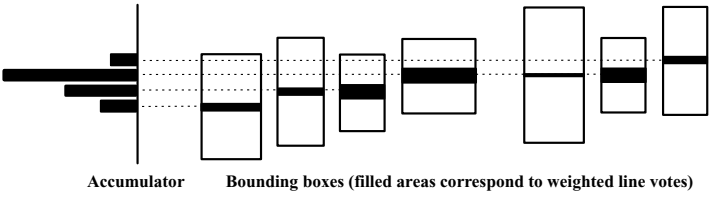
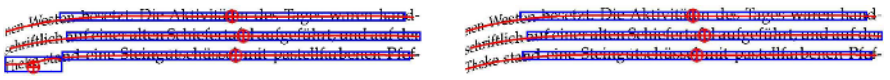


Fig. 3. Text line detection



(a) (b)

Fig. 4. Seed, core zone and approximated run of warped text lines before (a) and after (b) broken line correction

In regions of extreme line deformation it is possible that curved lines are broken by the detection algorithm (see Fig. 4 (a)). Therefore possibly broken line parts are melted provided that the extrapolated line run of at least one part hits the border component's bounding box of the adjacent line part.

The run of each distorted text line is then polynomially approximated. A k-th degree polynomial $y = a_0 + a_1 x + \cdots + a_k x^k$ can be determined using *least squares fitting*. Setting the partial derivatives $\partial E/\partial a_i$ of the residual $E = \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k} a_j x_i^j \right)^2$ to 0 leads to the following matrix equation which can be solved numerically:

$$\begin{pmatrix} n & \sum_{i=1}^{n} x_i & \cdots & \sum_{i=1}^{n} x_i^k \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \cdots & \sum_{i=1}^{n} x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_i^k & \sum_{i=1}^{n} x_i^{k+1} & \cdots & \sum_{i=1}^{n} x_i^{2k} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \vdots \\ \sum_{i=1}^{n} x_i^k y_i \end{pmatrix} \tag{1}$$

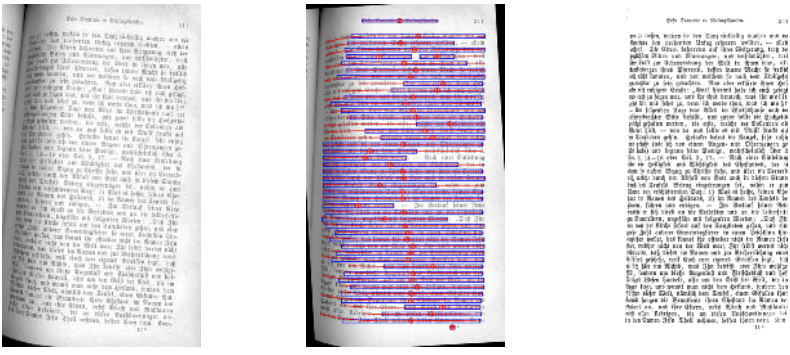
Note that we choose $k = \min(4, n-2)$ depending on how many centers have been localized.

The polynomial approximations of the text lines are now used to construct a dense source mesh. As the approximation is not reliable near the margins of short lines these grid values are substituted by averaging the run of surrounding text lines. The vertical grid lines are created by subdividing the lines into small segments of about H_{avg} width.

The target mesh can be assumed to be rectangular. Its vertical position is determined by the vertical position of the text block's non-distorted core zone which can be calculated from the detected text lines' core zones. Thus a rectified target (sub)image can now be created by applying any 2D-warping algorithm. For our tests we used a classical bilinear transformation.

4 OCR Results Comparison

We run our OCR tests twice before and after image binarization and restoration using the OCR software ABBYY FineReader 7.0. Like Zhang and Tan in [10] we use *precision* and *recall* as a measure for comparison of OCR improvement. Precision is hereby set to the quotient of the number of characters correctly detected and the total number of characters detected whereas recall denotes the quotient of the number of characters correctly detected and the total number of



(a) (b) (c)

Fig. 5. Skewed distorted (a), de-skewed (b), and binarized and rectified (c) sample fraktur document image

Table 1. OCR improvement (precision and recall)

	Original document		Bin. and rectified document	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Ave. of 50 (300 dpi)	92.81	88.99	97.53	96.29
Ave. of 50 (150 dpi)	87.61	74.41	88.81	80.70
Ave. of 10 skewed (300 dpi)	32.77	1.62	98.03	94.94

characters in the document. It was not possible to compare the OCR improvements to the results presented in [10] because the text base was not specified. Wu and Agam did not provide any OCR test results in [9]. Thus, we chose a set of 50 different sample documents containing a wide variety of different font sizes, typefaces and layouts. These documents were scanned from bound volumes at resolution 150 and 300 dpi, respectively. The orientation of the documents was varied in order to create different types of distortion. Note that at 150 dpi resolution OCR accuracy is higher for grayscale than for binary images which explains the low increase in precision for the rectified documents. Additionally we experimented with a set of 10 severely skewed warped documents (5 to 40 degrees). Without manual correction the OCR program detected only 6.41 percent of the documents' content. The results of our tests are presented in Table 1.

5 Conclusion and Future Development

A novel algorithm for image warping detection and correction is presented. The advantage of this approach is that few assumptions are required relating to the type of distortion that can be processed. The algorithm was implemented to be used in a recent digitization project and can be used without human interaction. Our tests show that processing distorted documents with this algorithm can

significantly improve the recognition accuracy of a downstream OCR engine. Future research will focus on the improvement of the correction of perspective deformation which may have multiply causes, for example variation of distance between the document and the scanner or camera.

Acknowledgements

The project *eCampus Duisburg* was supported by the German Federal Ministry of Education and Research (BMBF).

References

[1] Amin, A., Fischer, S., Parkinson, A. F., Shiu, R., *Comparative Study of Skew Detection Algorithms*, Jour. of Electronic Imaging SPIE, USA, 1996, pp. 443-451

[2] Biella, D., Dyllong, E., Kaiser, H., Luther, W., Mittmann, Th., *Edition électronique de la réception de Nietzsche des années 1865 à 1945*, Proc. ICHIM03, Paris, France, Sept. 2003

[3] Biella, D., Luther, W., *Mobile verteilte Dokumentenrecherche in Bibliotheken und Archiven*, In: INFORMATIK 2003 - Innovative Informatikanwendungen, Vol. 1, GI 2003, Germany, pp. 298-302

[4] Biella, D., Luther, W., Pilz, Th., *A web-based System for Assisted Literature Research*, In: Proceedings of the 3rd European Conference on e-Learning, ECEL 2004, Nov. 2004, Paris, France, pp. 15-24

[5] Cao, H., Ding, X., and Liu, C., *A Cylindrical Surface Model to Rectify the Bound Document Image*, Ninth IEEE ICCV 2003 Vol. 1, Nice, France, Oct. 2003, pp. 228-233

[6] Fletcher, L. A., Kasturi, R., *A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images*, IEEE Trans. Pattern Anal. Mach. Intell. 10(6), 1988, pp. 910-918

[7] Otsu, N., *A Threshold Selection Method from Graylevel Histograms*, IEEE Trans. Sys. Man Cybern. 9(1), 1979, pp. 62-66

[8] Savakis, A. E., *Adaptive Document Image Thresholding Using Foreground and Background Clustering*, Proc. of ICIP 1998, 1998, pp. 785-789

[9] Wu, C., Agam, G., *Document Image De-Warping for Text/Graphics Recognition*, Proc. of Joint IAPR 2002 and SPR 2002, Windsor, Ontario, Canada, Aug. 2002, pp. 348-357

[10] Zhang, Z., Tan, C. L., *Correcting Document Image Warping Based on Regression of Curved Text Lines*, ICDAR 2003, Aug. 2003, Edinburgh, UK, pp. 589-593

[11] Zhang, Z., Tan, C. L., Fan, L., *Estimation of 3D Shape of Warped Document Surface for Image Restoration*, ICPR 2004, Aug. 2004, Cambridge, UK, pp. 486-489

Order Independent Image Compositing

Conrad Bielski and Pierre Soille

Land Management Unit of the Institute for Environment and Sustainability
Joint Research Centre of the European Commission
T.P. 262, I-21020 Ispra (Va), Italy
{Conrad.Bielski,Pierre.Soille}@jrc.it

Abstract. Image compositing is defined as the assembling of two or more overlapping images into a single image. Recently, a morphological image compositing algorithm was proposed that automatically positions seam lines along salient image structures. This algorithm requires that all images used in the compositing procedure be held in computer memory. Hence, when composing large images such as satellite imagery acquired over a large region, the maximal capacity of random access memory of 32 bit computers is rapidly exceeded. In this paper, we present a parallel algorithm producing the same results whilst requiring only one input image at a time to be held in memory. The algorithm is illustrated for the automatic production of pan-European mosaics of Landsat images.

1 Introduction

Image mosaicing can be defined as the registration of two or more images which are then combined into a single image. This procedure is applied when more than one image or scene was acquired in order to cover the entire study area. This is a routine image processing task in remote sensing, astronomy, microscopy, medical imaging and, more recently, digital photography. Prior to image mosaicing, the scenes themselves must be geometrically registered, i.e. all images must be aligned geometrically so that differences between viewpoints and/or sensors are corrected. After geometric registration, the images are assembled so as to create a single scene having a field of view corresponding to the union of the field of views of the input scenes. The assembling of the individual registered images must address the problem of determining how the pixels belonging to regions visible from more than one scene (i.e., overlapping domains) should be represented. This problem is sometimes referred to as *image compositing* [1]. Seam lines that define the extent of the imagery used in image compositing in many cases are produced manually [2,3] while many remote sensing software solutions produce visible seams given the user-defined choice of which image should be on top, see Fig. 1.

Recently, the second author of this paper proposed a morphological [4] image compositing algorithm to automatically position seam lines along salient image structures [5]. The proposed morphological image compositing algorithm automatically adapts to the morphology of the image objects by following their

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1076–1083, 2005.

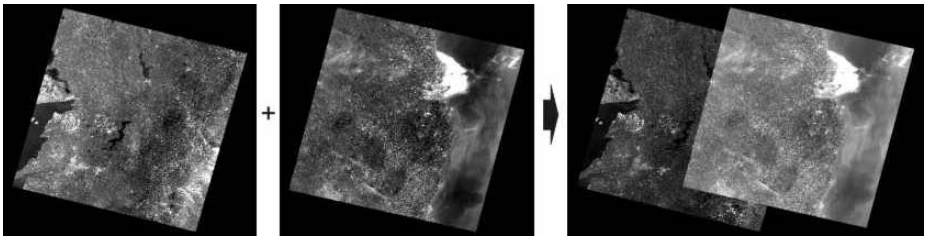


Fig. 1. Two co-registered Landsat TM scenes of Ireland are shown on the left. Basic compositing simply chooses which scene to place on top thereby creating a seam line that follows the data extent of the top image.

boundaries. This is achieved through marker controlled segmentation of the gradient intensity where the marker is the internal border of the overlapping regions and the gradient intensity is the point-wise minimum of gradients based on the overlapping input images. The number of overlapping images also drives the seam generation sequence because regions with fewer overlapping images must be processed before the seam line can continue to regions with a greater number of overlapping images.

The order independent image compositing algorithm presented in this paper was developed because the above described compositing technique could not be applied for generating very large mosaics. Indeed, it requires a single image of the size of the final mosaic to be held in memory which is impossible when dimensions exceed the available RAM size (4 GB for our system). When such processing is required in the field or on the fly (e.g., for web mapping services), portable or desktop devices cannot handle such large imagery. This problem is solved by proposing a scheme allowing for the processing of one input image at a time. However, this introduces the problem of order dependence since the final mosaic depends on the processing order of the input images. We address this problem by presenting an order independent algorithm. That is, the same results are obtained whatever sequence the images within the dataset are processed. Consequently, the resulting order independent algorithm is suitable for parallel processing. In Sec. 2, the method details are explained while Sec. 3 presents the application of this algorithm to Landsat Thematic Mapper (TM) imagery.

2 Methodology

Prior to image compositing, the input images must first be co-registered in a common reference coordinate system and resampled at the same spatial resolution. This is a usual procedure when analysing remotely sensed images where scenes are orthorectified, reprojected, and gridded at the node of a given sampling grid defined by a given projection of the spheroid on to the plane (unless geodetic coordinates with a given sampling increment along parallels and meridians are used). Gridding involves the application of an interpolation function such as that defined by the cubic convolution [6].

We assume that n input co-registered images denoted by $f_1, \ldots, f_n$ are available and that neither of them is a subset of the union of the others: for all $i \in \{1, \ldots, n\}$, $\mathcal{D}_i \not\subseteq \bigcup_{j \mid j \neq i} \mathcal{D}_j$, where $\mathcal{D}_i$ denotes the definition domain of image f_i. Indeed, in case the definition domain of an image falls within the union of all the other image definition domains, the former image will not contribute to the final result so that its inclusion in the input dataset is useless. We aim at defining a unique value for each pixel of the composed image f whose definition domain equals the union of all input definition domains: $\mathcal{D}_f = \cup_{i=1}^{n} \mathcal{D}_i$. The principle of the order independent compositing algorithm is first presented and then pseudo-code describing its implementation is provided.

2.1 Principle

Each input image f_i contains an arbitrary number of lines and columns corresponding to a rectangular frame. Unfortunately, the scene data extent does not necessarily correspond to the frame extent due to the acquisition mode and the orthorectification process, see example in Fig. 1. Therefore, $\mathcal{D}_i$ refers to the domain of definition of the ith image in the sense of the subset of the image frame where data values are present. Figure 2 (left side) shows a case with 4 input images where, for ease of representation, the rectangular frames match the image definition domains. We then construct an *overlap matrix*. This is a symmetrical $n \times n$ indicator matrix indicating whether the definition domains of an arbitrary image pair overlap or not:

$$m_{i,j} = \begin{cases} 1, \text{ if } \mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset, \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

For example, the overlap matrix corresponding to Fig. 2 (left side) is as follows:

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

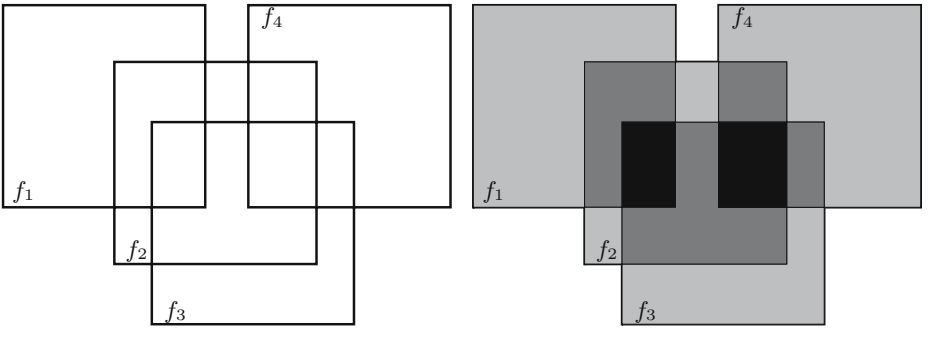


Fig. 2. Left: Four overlapping images. Right: the overlap levels ranging from 1 (light grey for no overlap) to 3 (black for 3 images overlapping).

In general, owing to the arbitrary shape of the definition domains $\mathcal{D}_i$, the overlap matrix cannot be constructed on the sole knowledge of the frame size and positioning of the upper left corner (or any other reference point) in the reference coordinate system. Hence, for all pairs of images whose frames intersect, we need to compute the intersection between their actual definition domains to assess whether these intersect or not.

Beyond the information available through the overlap matrix, the number of overlapping images (called *overlap level* hereafter) for any given pixel must be known. Indeed, seam lines are detected in an ordered fashion, starting with regions with the least overlap level greater than 1 and proceeding to the subsequent level until the maximum number of overlap is reached. For example, Fig. 2 (right side) shows the domains of equal overlapping levels with a specific grey shade: light grey for overlap level equal to 1 (no overlap), dark grey for overlap level 2, and black for overlap level 3.

We then proceed as follows. The overlap matrix is scanned row by row. The index of the row defines the current *anchor image*. Actual processing is restricted to the region defined by its frame (we assume that the definition domain is buffered by the image frame). Within this region, the least overlap level greater than 1 defines the *current overlap level* denoted by h_{crt}. We also track whether an overlap level higher than the current level occurs. This is used to determine whether the overlap matrix needs to be scanned again later on. The morphological compositing routine [5] is then called while restricting its effect to the processing of those regions whose overlap level is equal to h_{crt}. The routine assigns each pixel of these regions to a unique source image so that the definition domain of the anchor image and those of the images intersecting it can be updated accordingly. This update can only remove some parts of these input definition domains since it concerns regions where more than one image was competing for the same domain. Order independence is achieved because identical results are obtained whatever order the anchor images are processed.

2.2 Implementation

In essence, the above principle is not too complex, however its implementation requires data to be carefully updated on the fly. Information with regards to the anchor image overlap level and changes to the definition domain with increasing overlap level needs to be taken into account because these changes affect the seam line generation.

Code implementation was divided into two steps: generating the overlap matrix and processing the overlap matrix. The overlap matrix is generated based on the list of available images to be mosaiced which can be of any length. List order determines the location of the anchor image within the overlap matrix, i.e., image list item 2 will be anchor image f_2 within the overlap matrix.

The second step involves the actual compositing of the mosaic based on the overlap matrix. Processing of anchor and overlap images follows each matrix row. Each ith matrix row was linked to the ith anchor image thus when dealing with the ith row it should also be understood that one is dealing with f_i as anchor

image. An auxiliary image g whose frame equals that of the current anchor image is used to calculate the current overlap level h_{crt}. We also assume that the actual definition domains are stored on disk as binary images where a pixel value of 1 indicates whether a pixel belongs to it. Pseudo-code for the order independent compositing is summarised hereafter:

1. DO
2. flag $\leftarrow$ false
3. FOR $i \leftarrow 1$ to n // scan the overlap matrix row by row
4. $g = \sum_{\{j \mid m_{i,j}=1\}} \mathcal{D}_j$ // pixel-wise addition (restricted to domain $\mathcal{D}_i$)
5. $h_{\max} = \max_{\mathbf{x}} g(\mathbf{x})$
6. IF $h_{\max} = 1$ THEN GOTO 3 // no more overlap
7. $h_{\text{crt}} \leftarrow \min_{\{\mathbf{x} \mid g(\mathbf{x})>1\}} g(\mathbf{x})$
8. IF $h_{\max} > h_{\text{crt}}$ THEN flag $\leftarrow$ true END IF
9. apply image compositing [5] to anchor image frame for overlap level $= h_{\text{crt}}$
10. update definition domains of all images such that $m_{i,j} = 1$
11. END FOR
12. WHILE flag = true

The implementation recursively goes through all overlap matrix rows until there are no more overlapping images. This is necessary because the current overlap level h_{crt} is unknown *a priori* and therefore must be evaluated each time an overlap matrix row is being processed (line 7). Note that different anchor images will have different overlap levels that are not necessarily sequential.

Success of this order independent strategy depends on the precise updating of the image definition domains. Updated image definition domains are based on two results: the image compositing procedure for the current overlap level and the re-insertion of areas with overlap levels greater than the current level. For the anchor image, the image compositing results always fall within its own definition domain and overlap levels greater than the current overlap level are re-inserted as required. The overlap images, on the other hand, can only be updated within the definition domain of the anchor image because information with respect to image definition domain overlap are only known within the definition domain of the anchor image for any given overlap matrix row. This will be more evident with the example shown in the next section. When the flag is not reset to true when scanning the overlap matrix (line 8), image compositing has been applied to the entire mosaic and the program stops.

3 Application

A sample application of the order independent image compositing procedure is presented based on Landsat Thematic Mapper (TM) imagery over the island of Sardegna, Italy. The imagery was taken from the Image2000 [7] dataset which covers all European Union member states. These images were acquired during the summer season over the 1999 to 2001 time frame. The island of Sardegna was chosen as the sample region to clearly present the results because only

five Landsat TM images are required to cover the entire island. However, the algorithm can be handle any number of scenes. For example, the composition of the 39 Landsat TM images covering Italy by the order independent image compositing procedure was obtained with no additional random access memory requirements.

The overlap matrix for Sardegna is presented in figure 3. Note that Landsat scenes are in no particular spatial order and overlap between definition domains is readily apparent: anchor images f_2 and f_3 have 2 images with overlapping definition domains, anchor images f_1 and f_4 have 3 images with overlapping definition domains and anchor image f_5 has 4 images with overlapping definition domains. Image compositing can only proceed within the image definition domain which is the region where actual remote sensing measurements are available. For the orthorectified Landsat TM scenes, this region is the rotated square within the image. Based on the above overlap matrix, image compositing was

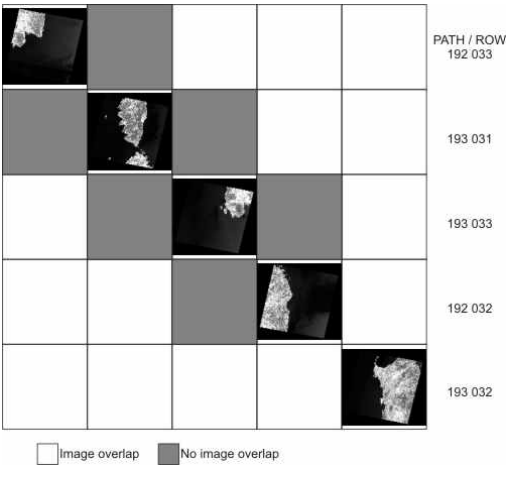


Fig. 3. Overlap matrix for the island of Sardegna, Italy. Five Landsat TM images cover this region (image path and row are presented to the right of the matrix) with band 5 (near infrared) presented in the diagonal.

applied. The step-by-step results of the order independent procedure are presented in figure 4. On the left is presented the actual number of image overlaps for this dataset. It shows that the greatest overlap level is 3 (white regions). During the order independent procedure, the dark grey regions (overlap level= 1) will never be processed because there is only a single image available in those areas. The light grey regions indicate areas where the overlap level is equal to 2. These regions must first be processed before image compositing can occur for regions where there are 3 images overlapping (i.e. overlap level = 3). This figure is only shown as a reference because in practice the overlap level is computed on the fly based on the images found within any given overlap matrix row thus ensuring order independence. The middle diagram in figure 4 presents the updated image definition domains after image compositing has been applied to all regions where the overlap level is 2. From the overlap level mosaic there should be three regions that persist in the updated image definition domains. These

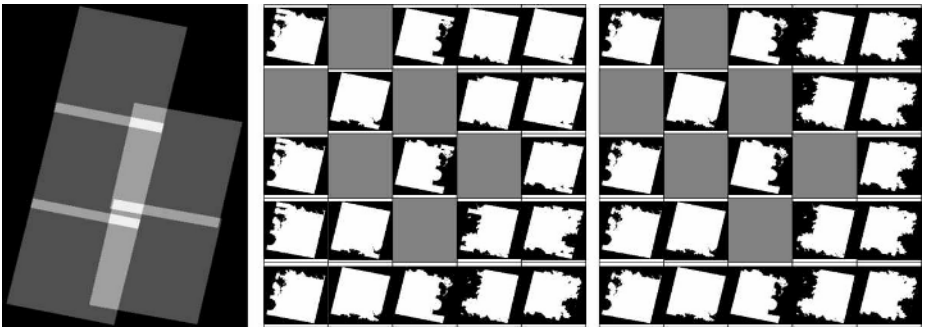


Fig. 4. Left: overlap level map for the Sardegna dataset. Middle: updated image definition domains for image overlap level 2. Right: final image definition domains, i.e., when the maximum overlap level which is equal to 3 is reached.

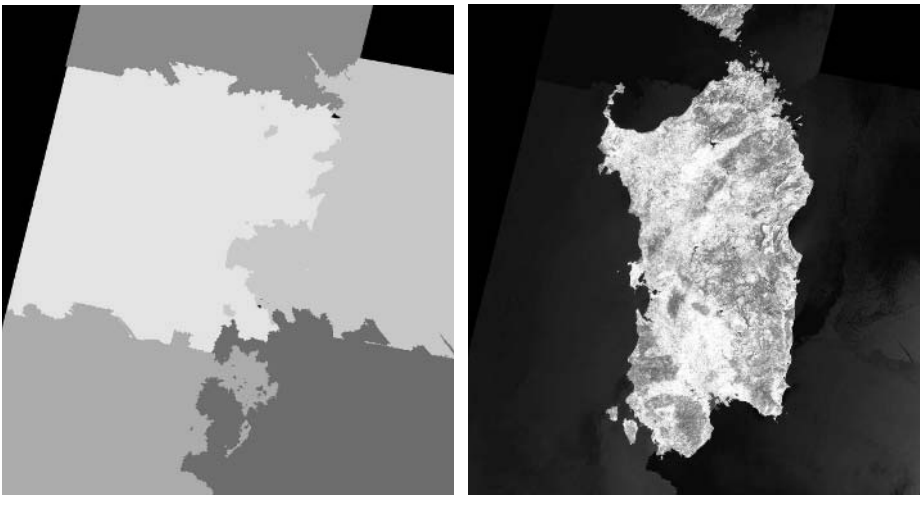


Fig. 5. The result of applying the order independent image compositing procedure. On the left is the mosaic of the final image definition domains and on the right is the mosaic based on Landsat TM band 5 imagery.

rectangular regions where the overlap level is greater than 2 can be seen in the corners of the images because they persist throughout the processing of overlap level 2. Note however that in the final row, $\mathcal{D}_{4,4}$ has been completely updated. This result was already mentioned in the principle section and occurs because all regions related the last row of the overlap matrix have already been processed. The rightmost diagram presents the final image definition domains. In this case, all definition domains fit perfectly together like a puzzle and therefore there is no more image overlap anywhere within the mosaic: $D_f = \cup_{i=1}^{n} \mathcal{D}_i$ and for all $i \neq j$, $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ since the $\mathcal{D}_i$'s have been updated through image composition. The final image definition domains are then used to create the image

mosaic presented in figure 5. Inland seam lines are barely visible even though no radiometric equalisation was performed between the different scenes taken at different times. In the sea however, seam lines are noticeable because of the lack of salient structures. Automatic compositing in these homogeneous regions is therefore primarily driven by low frequency fluctuations caused by variable atmospheric and sea conditions at the time of acquisition.

4 Conclusion and Perspectives

Efficient order independent image compositing of large images can be achieved based on the proposed algorithm. In addition, the algorithm enables the user to add further functionalities such as the minimisation of specific structures occurring within the overlap regions. For example, cloud and shadow complexes were removed when composing the satellite images shown in this paper. For conciseness this was not presented here but will be detailed in an extended version. The algorithm also directly applies to 3-dimensional imagery such as those produced by imaging tomography devices. Given the size of the 3-D input images, order independent processing on an image by image basis is an asset.

References

1. Porter, T., Duff, T.: Compositing digital images. Computer Graphics 18 (1984)
2. Hummel-Miller, S.: A digital mosaicking algorithm allowing for an irregular join line. Photogrammetric Engineering and Remote Sensing 55 (1989) 43–47
3. Nunes de Lima, V., ed.: IMAGE 2000 and CORINE Land Cover 2000 —Products and Methods—. European Commission, Joint Research Centre (2005) In Press.
4. Soille, P.: Morphological Image Analysis: Principles and Applications. corrected 2nd printing of the 2nd edn. Springer-Verlag, Berlin and New York (2004)
5. Soille, P.: Morphological compositing. IEEE Transactions on Pattern Analysis and Machine Intelligence (2005) [Submitted].
6. Keys, R.: Cubic convolution interpolation for digital image processing. IEEE Transactions on Acoustics, Speech, and Signal Processing 29 (1981) 1153–1160
7. Nunes de Lima, V., Peedell, S.: IMAGE2000-The European spatial reference to support environmental information. In: Proceedings of 18th International Conference Informatics for Environmental Protection (EnviroInfo), Geneva, 21–23 October. Volume II., Genèeve, Editions du Tricorne (2004) 268–276 [Invited session].

Improving SIFT-Based Object Recognition for Robot Applications*

Patricio Loncomilla[1,2] and Javier Ruiz-del-Solar[1,2]

[1] Department of Electrical Engineering, Universidad de Chile
[2] Center for Web Research, Department of Computer Science, Universidad de Chile
{ploncomi, jruizd}@ing.uchile.cl

Abstract. In this article we proposed an improved SIFT-based object recognition methodology for robot applications. This methodology is employed for implementing a robot-head detection system, which is the main component of a robot gaze direction determination system. Gaze direction determination of robots is an important ability to be developed. It can be used for enhancing cooperative and competitive skills in situations where the robots interacting abilities are important, as for example, robot soccer. Experimental results of the implemented robot-head detection system are presented.

1 Introduction

Object recognition algorithms based on scale and orientation invariant local descriptors have experienced and impressive development in the last years ([1][4][5]). Most successful proposed systems employ either the Harris detector [3] or SIFT (Scale Invariant Feature Transform) features [1] as building blocks. Object recognition systems based on SIFT features have shown a higher robustness and stability than those based on the Harris detector [1]. They have been used for building diverse kind of applications (object recognition, image alignment, robot localization, etc.), however, they have almost not been used for robot or robot parts recognition.

On the other hand, gaze direction determination between robots can be used for enhancing cooperative and competitive skills in situations where the robots interacting abilities are important. For instance, in robot soccer, gaze direction determination of opponents and teammates is a very important ability for anticipating the others' behavior. However, this ability is still not developed. We aim at reverting this situation by proposing a gaze direction determination system for robots, based on SIFT features [8]. In this approach, gaze direction determination is based on a robot-head pose detection system, which employs two main processing stages. In the first stage, scale and orientation invariant local descriptors of the observed scene are computed. Then, in the second stage these descriptors are matched against descriptors of robot-head prototypes already stored in a model database. After the robot-head pose is recognized, the robot gaze direction is determined using a head model of the

* This research was funded by Millenium Nucleus Center for Web Research, Grant P04-067-F, Chile.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1084 – 1092, 2005.

observed robot, and the current 3D position of the observing robot camera. (In the employed robots (Sony AIBO) the relation between head and camera pose is fixed, therefore it is not required additional camera pose determination.)

While developing this robot-head pose detection system, we realize that due to the physical characteristics of some robots models such as the SONY AIBO ERS7 (rounded head shape and poor-textured head surface producing a high amount of highlights) and the small size of the AIBO camera images (208x160), it is very difficult to obtain reliable SIFTs on them. Therefore, the traditional SIFT computation and matching algorithms do not work very well here. For this reason, we had the necessity of improving these algorithms to robustly reject false detections.

The main objective of this paper is to propose an improved SIFT-based object recognition system. The local descriptors computation and matching are based on [1], but many important parts of the method have been improved for fitting it to the robot-head detection problem, and for maintain detection accuracy while incrementing the number of keypoint matches. Experimental results consisting on the application of the developed methodology to the robot-head detection problem are shown.

2 Improved SIFT-Based Object Recognition

2.1 Scale-Invariant Local Descriptors Computation

Detection of Scale-Space Extrema. A difference-of-Gaussian (DoG) function is employed for identifying potential interest points that are invariant to scale and orientation. These keypoints are searched over all scales and image locations using a fast scale-space transformation, starting with a small $\sigma = 0.8$ scale level and no image duplication. It can be proved that by using the DoG over the scale-space, image locations that are invariant to scales can be found, and that these features are more stable than other computed using the gradient, Hessian or Harris corner function [1]. The scale-space of an image is defined as a function, $L(x,y,\sigma)$, which corresponds to the convolution of the image with a Gaussian of scale σ. The DoG function between two nearby scales separated by a constant multiplicative factor *k* can be computed as:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

The local extrema (maxima and minima) of $L(x,y,\sigma)$ are detected by comparing each sample (x,y,σ) with its 26 neighbors in the scale-space (8 in the same scale, 9 in the scale above and 9 in the scale below).

Accurate Keypoint Localization. The detected local extrema are good candidates to become keypoints, but previously they need to be exactly localized. Subsequently, local extrema with low contrast are rejected because they are sensitive to noise, and keypoints that correspond to edges are also discarded.

First, local extrema to sub-pixel / sub-scale accuracy are found by fitting a 3D quadratic to the scale-space local sample point. The quadratic function is computed using a second order Taylor expansion having the origin at the sample point [2]:

$$D(\mathbf{x}) = D(\mathbf{0}) + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{1}$$

where **x** is the offset from the sample point. Then, by taking the derivate with respect to **x** and setting it to zero, the location of the extrema of this function is given by:

$$\hat{\mathbf{x}} = -H^{-1}\nabla D(\mathbf{0}) \tag{2}$$

In [1][2] the Hessian and gradient are approximated by using differences of neighbor samples points. The problem with this coarse approximation is that just 3 samples are available in each dimension for computing the Hessian and gradient using pixel differences, which produces a non-accurate result. We improve this computation by using a real 3D quadratic approximation of the scale-space, instead of discrete pixel differences. Our 3D quadratic approximation function is given by:

$$\tilde{D}(x, y, \sigma) = a_1 x^2 + a_2 y^2 + a_3 \sigma^2 + a_4 xy + a_5 x\sigma + a_6 y\sigma + a_7 x + a_8 y + a_9 \sigma + a_{10}$$

Using the 27 samples contained in the 3x3x3 cube under analysis, the unknowns (a_i) can be found. Using vector notation, this linear system will be given by:

$$\begin{bmatrix} x_1^2 & y_1^2 & \sigma_1^2 & x_1 y_1 & x_1\sigma_1 & y_1\sigma_1 & x_1 & y_1 & \sigma_1 & 1 \\ x_2^2 & y_2^2 & \sigma_2^2 & x_2 y_2 & x_2\sigma_2 & y_2\sigma_2 & x_2 & y_2 & \sigma_2 & 1 \\ & & & & \cdots & & & & & \\ x_{27}^2 & y_{27}^2 & \sigma_{27}^2 & x_{27} y_{27} & x_{27}\sigma_{27} & y_{27}\sigma_{27} & x_{27} & y_{27} & \sigma_{27} & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdots \\ a_{10} \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ \cdots \\ D_{27} \end{bmatrix}$$

where D_i corresponds to the sample point value (intensity) i. We can write this linear system as $\mathbf{Ba} = \mathbf{d}$. The least-squares solution for the parameters **a** is given by:

$$\mathbf{a} = \left(\mathbf{B}^\mathbf{T}\mathbf{B}\right)^{-1}\mathbf{B}^\mathbf{T}\mathbf{d}$$

It should be stressed that the matrix $\left(\mathbf{B}^\mathbf{T}\mathbf{B}\right)^{-1}\mathbf{B}^\mathbf{T}$ needs to be computed once for the whole image, and that it can be eventually pre-computed. Now, the accurate location of the extrema can be computed using (2), with the following Hessian and gradient expression:

$$\mathbf{H} = \begin{bmatrix} 2a_1 & a_4 & a_5 \\ a_4 & 2a_2 & a_6 \\ a_5 & a_6 & 2a_3 \end{bmatrix};\ \nabla\tilde{D}(0) = \begin{bmatrix} a_7 \\ a_8 \\ a_9 \end{bmatrix} \tag{3}$$

Second, local extrema with a contrast lower than a given threshold Th_{contr}, are discarded ($\left|\tilde{D}(\hat{\mathbf{x}})\right| < Th_{contr}$).

Third, extrema corresponding to edges are discarded using curvature analysis. A peak that corresponds to an edge will have a large principal curvature across the edge but a small one in the perpendicular direction. The curvature can be computed from the 2x2 submatrix $\mathbf{H}_{xy}$ that considers only the x and y components of the Hessian. Taking into account that we are interested on the ratio between the eigenvalues, we will discard extrema in which the ratio of principal curves is above a threshold r, or equivalently local extrema that fulfill the following condition (see [3] for a deeper explanation):

$$\frac{\mathrm{Tr}(\mathbf{H}_{xy})^2}{\mathrm{Det}(\mathbf{H}_{xy})} > \frac{(r+1)^2}{r}$$

In [1] $\mathbf{H}_{xy}$ is computed be taking differences of neighbor sample points. As already mentioned, this approximation produces a non-accurate result. We improved this situation by computing $\mathbf{H}_{xy}$ from (3).

Orientation Assignment. By assigning a coherent orientation to each keypoint, the keypoint descriptor can be represented relative to this orientation and hence achieve invariance against rotations. The scale of the keypoint is employed for selecting the smoothed image L(x,y) with the closest scale, and then the gradient magnitude and orientation are computed as:

$$\mathrm{m}(x,y) = \sqrt{(L(x+1,y)-L(x-1,y))^2 + (L(x,y+1)-L(x,y-1))^2}$$
$$\theta(x,y) = \tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y)))$$

As in [1], an orientation histogram is computed from the gradient orientations at sample points around the keypoint (*b1* bins are employed). A circular Gaussian window whose size depends of the scale of the keypoints is employed for weighting the samples. Samples are also weighted by its gradient magnitude. Then, peaks in the orientation histogram are detected: the highest peak and peaks with amplitudes within 80% of the highest peak. Orientations corresponding to each detected peak are employed for creating a keypoint with this orientation. Hence, multiple keypoints with the same location and scale but different orientation can be created (empirically, about 85% of keypoints have just one orientation).

Keypoint Descriptor Computation. For each obtained keypoint, a descriptor or feature vector that considers the gradient values around the keypoint is computed. The obtained descriptors are invariant against some levels of change in 3D viewpoint and illumination. The keypoints and their associated descriptors are knows as SIFT (Scale Invariant Feature Transform) features or just SIFTs.

First, in the keypoint scale the gradient magnitude and orientation are computed around the keypoint position (usually a neighborhood of 8x8 or 16x16 pixels is considered). Then, the gradient magnitudes are weighted by a Gaussian window, and the coordinates of the descriptor as well as the gradient orientations are rotated relative to the keypoint orientation. Second, the obtained gradient values are accumulated into orientation histograms summarizing the contents of 4x4 subregions (*b2* bins are employed). Thus, a descriptor vector is built, where each vector component is given by an orientation histogram. Depending on the neighborhood size, 2x2 or 4x4 vectors are obtained. Third, illumination effects are reduced by normalizing the descriptors' vector to unit length. Abrupt brightness changes are controlled by limiting the intensity value of each component of the normalized vector. Finally, descriptors vectors are re-normalized to unit length.

2.2 Matching of Local Descriptors and Object Prototypes Descriptors

The matching process consists of nine processing stages. In the first stage, the image keypoint descriptors are individually matched against prototype descriptors. In the second stage this matching information is employed for obtaining a coarse prediction of the object pose. In the third stage possible affine transformations between a prototype and the located object are determined. In the later six stages these affine

transformations are verified, and some of them discarded or merged. Finally, if the object is present in the image just one affine transformation should remain. This transformation determines the object pose. In the original work of Lowe [1], only the first four stages here employed were considered. The five additional verification stages improve the detection accuracy.

Individual Keypoint Descriptors Matching. The best candidate match for each image keypoint is found by computing its Euclidian distance with all keypoints stored in the database. It should be remembered that each prototype includes several keypoint descriptors. Considering that not all keypoints are always detected (changes in illumination, pose, noise, etc.) and that some keypoints arise from the image background and from other objects, false matches should be eliminated. A first alternative is to impose a minimal value to a match to be considered correct. This approach has proved to be not robust enough. A second alternative consists on comparing the distance to the closest neighbor to that of the second-closest neighbor. If this ratio is greater than a given threshold, it means than this image keypoint descriptor is not discriminative enough, and therefore discarded. In [1] the closest neighbor and second-closest neighbor should come from a different object model (prototype). In the current case this is not a good idea, because we have multiple views of the same object (e.g. a robot). Therefore, we allow that the second-closest neighbor can come from the same prototype than the closest neighbor. The image under analysis as well as the prototype images generates a lot of keypoints, hence having an efficient algorithm for computing the keypoint descriptors distance is a key issue. This nearest neighbor indexing is implemented using the Best-Bin-First algorithm [6], which employs a k-d tree data structure.

Object Pose Prediction. In the pose space a Hough transform is employed for obtaining a coarse prediction of the object pose, by using each matched keypoint for voting for all object pose that are consistent with the keypoint. A candidate object pose is obtained if at least 3 entries are found in a Hough bin. Usually, several possible object pose are found. The prediction is coarse because the similarity function implied by the four parameters (2D location, orientation and scale) is only an approximation of the 6 degree-of-freedom of a 3D object. Moreover, the similarity function cannot account for non-rigid deformations.

Finding Affine Transformations. In this stage already obtained object pose are subject to geometric verification. A least-squares procedure is employed for finding an affine transformation that correctly account for each obtained pose. An affine transformation of a prototype keypoint (x,y) to an image keypoint (u,v) is given by:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

where the m_i represent the rotation, scale and stretch parameters, and t_x and t_y the translation parameters. The parameters can be found if three or more matched keypoints are available. Using vector notation, this linear system will be given by:

$$\begin{pmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \dots & & & \\ & & \dots & & & \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} u \\ v \\ \dots \\ \dots \end{pmatrix}$$

We can write this linear system as $\mathbf{Cp} = \mathbf{u}$. Finally, the least-squares solution for the parameters $\mathbf{p}$ is given by:

$$\mathbf{p} = \left(\mathbf{C}^{\mathrm{T}}\mathbf{C}\right)^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{u}.$$

Affine Transformations Verification Using a Probabilistic Model. The obtained model hypotheses, i.e. affine transformations, are subject to verification using a probabilistic model to help to reject false detections (see detailed description in [7]).

Affine Transformations Verification Based on Geometrical Distortion. A correct detection's affine transformation shouldn't deform very much an object when mapping it. Given that we have just a hypothesis of the object pose, it is not easy to determine the object distortion. However, we do have the mapping function, i.e. the affine transformation. Therefore, we can verify if the mapping function produce distortion or not using a known, regular and simple object, such as a square. The affine transformation of a square should produce a rotated parallelogram. If the affine transformation does not produce a large distortion, the conditions that the transformed object should fulfill are (see notation in fig. 1):

$$\max\left\{\frac{d(AB)/d(A'B')}{d(BC)/d(B'C')}, \frac{d(BC)/d(B'C')}{d(AB)/d(A'B')}\right\} < th_{prop} \;;\; \alpha = \sin^{-1}\left|\frac{\det(\overrightarrow{A'B'} \quad \overrightarrow{B'C'})}{d(\mathrm{A'B'}) \times d(\mathrm{B'C'})}\right| > th_{\alpha}$$

$\overrightarrow{A'B'}$ is a vector from *A'* to *B'*, $\det\left(\overrightarrow{A'B'} \quad \overrightarrow{B'C'}\right)$ computes the parallelogram area.

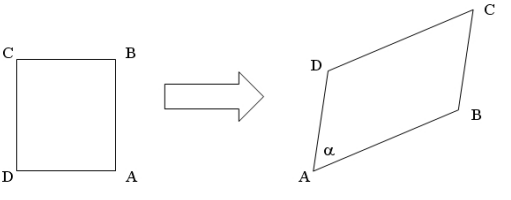


Fig. 1. Affine mapping of a square into a parallelogram

Affine Transformations Verification Based on Spatial Correlation. Affine transformations producing low lineal correlation, r_s, between the spatial coordinates of the matched SIFTs in the image and in the prototype are discarded:

$$r_s = \min\left(\max(r_{xx}, r_{xy}), \max(r_{yx}, r_{yy})\right) < th_{rs}$$

r_{xx} and r_{yy} correspond to the correlation in the x and y directions of the *N* matched SIFTs, while r_{xy}=r_{yx} corresponds to the cross correlation between both directions. r_{xx} and r_{xy} are calculated as (r_{yy} and r_{yx} are computed in a similar way):

$$r_{xx} = \left| \frac{\sum_{i=1}^{N} \left(x_i - \overline{x}\right)\left(x'_i - \overline{x'}\right)}{\sqrt{\sum_{i=1}^{N} \left(x_i - \overline{x}\right)^2 \sum_{i=1}^{N} \left(x'_i - \overline{x'}\right)^2}} \right| ; \; r_{xy} = \left| \frac{\sum_{i=1}^{N} \left(x_i - \overline{x}\right)\left(y'_i - \overline{y'}\right)}{\sqrt{\sum_{i=1}^{N} \left(x_i - \overline{x}\right)^2 \sum_{i=1}^{N} \left(y'_i - \overline{y'}\right)^2}} \right|$$

Affine Transformations Verification Based on Graphical Correlation. Affine transformations producing low graphical correlation, r_g, between the object prototype image and the candidate object subimage can be discarded:

$$r_g = \frac{\sum_{u=0}^{U} \sum_{v=0}^{V} \left(I(u,v) - \overline{I}\right)\left(I'\left(x_{TR}(u,v), y_{TR}(u,v)\right) - \overline{I'}\right)}{\sqrt{\sum_{u=0}^{U} \sum_{v=0}^{V} \left(I(u,v) - \overline{I}\right)^2 \sum_{u=0}^{U} \sum_{v=0}^{V} \left(I'\left(x_{TR}(u,v), y_{TR}(u,v)\right) - \overline{I'}\right)^2}} < th_{rg}$$

The affine transformation is given by $\{x=x_{TR}(u,v),\ y=y_{TR}(u,v)\}$. $I(u,v)$ and $I'(x,y)$ correspond to the prototype image and the candidate object subimage, respectively.

Affine Transformations Verification Based on the Object Rotation. In some real-world situations, real objects can have restrictions in the rotation (respect to the body plane) they can suffer. For example the probability that a real robot is rotated in 180° (inverted) is very low. For a certain affine transformation, the rotation of a detected object with respect to a certain prototype can be determined using the SIFTs keypoint orientation information. Thus, the object rotation, *rot*, is computed as the mean value of the differences between the orientation of each matched SIFTs keypoint in the prototype and the corresponding matched SIFTs keypoint in the image. Transformations producing large *rot* values can be discarded ($rot > th_{rot}$).

Affine Transformations Merging Based on Geometrical Overlapping. Sometimes more than one correct affine transformation corresponding to the same object can be obtained. There are many reasons for that, small changes in the object view respect to the prototypes views, transformations obtained when matching parts of the object as well as the whole object, etc. When these multiple, overlapping transformations are detected, they should be merged. As in the case when we verify the geometrical distortion produce by a transformation, we perform a test consisting in the mapping of a square by the two candidate affine transformations to be joined. The criterion for joining them is the overlap, *over*, of the two obtained parallelograms (see notation in fig. 1):

$$over = 1 - \frac{dist(A'_1 A'_2) + dist(B'_1 B'_2) + dist(C'_1 C'_2) + dist(D'_1 D'_2)}{perimeter(A'_1 B'_1 C'_1 D'_1) + perimeter(A'_2 B'_2 C'_2 D'_2)} > th_{over}$$

It should be also verified if the difference between the rotations produced for each transform is not very large. Therefore, two transforms will be joined if:

$$\left| rot_1 - rot_2 \right| < th_{diff\_rot}$$

3 Robot-Head Pose Detection

Basically, the robot-head pose is determined by matching image descriptors with descriptors corresponding to robot-head prototype images already stored in a model database. The employed prototypes correspond to different views of a robot head, in our case the head of an AIBO ERS7 robot. Because of in the context of the RoboCup four-legged league, we are interested on recognizing the robot pose as well as the robot identity (number); prototypes for each of the four players are stored in the database. In figure 2 are displayed the 16 prototype heads corresponding to one of the robots. The pictures were taken every 22.5°.

4 Experimental Results and Analysis

Robot-head detection experiments using real-world images were performed. In all of these experiments the 16 prototypes of robot player "1" were employed (see fig. 2). A database consisting on 39 images taken on a four-legged soccer field was built. In these images robot "1" appears 25 times, and other robots appear 9 times. 10 images contained no robots at all. In table 1 are summarized the obtained results. If we consider full detections, in which both, the robot-head pose as well as the robot identity is detected, a detection rate of 68% is obtained. When we considered partial detections, i.e. only the robot identity is determined, a detection rate of 12% is obtained. The combined detection rate is 80% while the number of false positives is

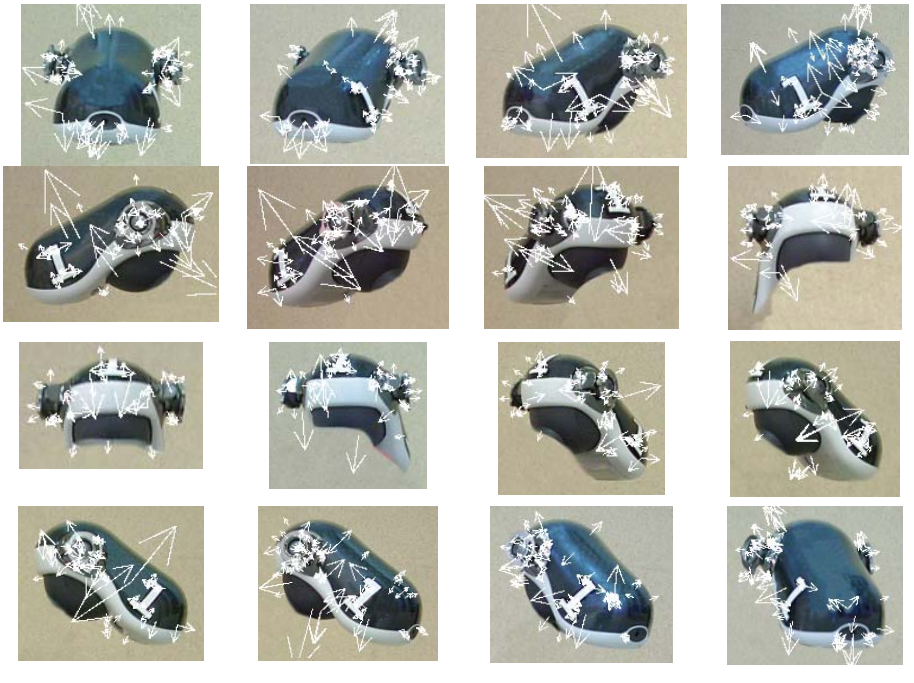


Fig. 2. AIBO ERS7 robot-head prototypes with their SIFTs. Pictures taken every 22.5°

Table 1. Robot-head detection of robot #1 (only robot #1 prototype were employed)

Full detections (head + identifier number)	17/25	68%
Partial detections(only the identifier number)	3/25	12%
Full + partial detections	20/25	80%
Number of false detections in 39 images		6

very low, just 6 in 39 images. These figures are very good, because when processing video sequences, the opponent or teammates robots are seen in several consecutive frames. Therefore, a detection rate of 80% in single images should be high enough for detecting the robot-head in few frames as an AIBO robot processes each frame in around 1 second.

We know that more intensive experiments should be performed for characterizing our system. Currently we are carrying out this characterization using a larger database (this database together with the robot prototypes database will be made public soon). However, we believe that these preliminary experiments show the high potential of the proposed methodology.

References

1. D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *Int. Journal of Computer Vision*, 60 (2): 91-110, Nov. 2004.
2. M. Brown and D. G. Lowe, Invariant Features from Interest Point Groups, *British Machine Vision Conference* - BMVC 2002, 656 – 665, Cardiff, Wales, Sept. 2002.
3. C. Harris and M. Stephens, A combined corner and edge detector, *Proc. 4th Alvey Vision Conf.*, 147-151, Manchester, UK, 1988.
4. F. Schaffalitzky and A. Zisserman, Automated location matching in movies, *Computer Vision and Image Understanding* Vol. 92, Issue 2-3, 236 – 264, Nov./Dec. 2003.
5. K. Mikolajczyk and C. Schmid, Scale & Affine Invariant Interest Point Detectors, *Int. Journal of Computer Vision*, 60 (1): 63 - 96, Oct. 2004.
6. J. Beis and D.G. Lowe, Shape Indexing Using Approximate Nearest-Neighbor Search in High-Dimensional Spaces, *Proc. IEEE Conf. Comp. Vision Patt. Recog,* 1000-1006, 1997.
7. D.G. Lowe, Local Features View Clustering for 3D Object Recognition, *Proc. of the IEEE Conf. on Comp. Vision and Patt. Recog*., 682 – 688, Hawai, Dic. 2001.
8. Loncomilla, and Ruiz-del-Solar (2005). Gaze Direction Determination of Opponents and Teammates in Robot Soccer, *RoboCup Symposium 2005*, Osaka, Japan, July 2005 (accepted).

Environment Topological Structure Recognition for Robot Navigation

Enver Sangineto[1] and Marco R. Iarusso[1,2]

[1] Centro di Ricerca in Matematica Pura ed Applicata (CRMPA),
[2] Dipartimento di Informatica e Automazione (DIA),
Universitá Roma 3, via della Vasca Navale 79, 00146, Rome, Italy

Abstract. Robot navigation using only abstract, topological information on the environment is strongly related to the possibility for a robot to unambiguously match information coming from its sensors with the basic elements of the environment. In this paper we present an approach to this challenging problem based on the direct recognition of the topological structure of the environment.

1 Motivations and Goals

In this paper we face the problem of robot navigation in an unknown indoor environment using information from an uncalibrated, single camera. In [10] robot navigation systems are categorized in systems based either on topological information (*topological maps*) or on metric information (*metric maps*). In the last two decades Computer Vision techniques have been frequently involved in the robot's self localization process, especially in conjunction with metric, a priori information. A typical vision-guided navigation system aims to recognize specific *landmarks* [2] whose exact position in the environment is known to the robot and allows it to self-localize. Some early systems use *artificial landmarks*, i.e., objects placed ad hoc in the environment for the recognition purpose, whose visual features (e.g., shape or color) facilitate their recognition. Examples of artificial landmarks are simple geometrical patterns [4] or bar codes [6].

On the other hand, *natural landmarks* can be any kind of objects belonging to the environment such as (in the indoor example): doors, windows, bulletin boards, etc.. Natural landmarks are less intrusive but usually more difficult to recognize. Dulimarta and Jain [5] use ceiling lights and door number plates as landmarks for self-localization. Kosaka and Kak [8] compare previously (handmade) 3D CAD models of the hallway environment with the input frames. Modeling with a Kalman filter the uncertainties in the position of the robot (influenced by the motion parameters) it provides a framework for the selection of the most likely positions of the landmarks in the image. A projective invariant technique is used by Wang and colleagues [9] for the robot self-localization task. Intersection points between straight vertical lines representing doors and the floor are used as features. A data base containing the projective invariants computed for the whole hallway is off-line built. On-line, the extraction of the intersection points

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1093–1100, 2005.

from the camera frame is used to search for in the data base and to compute the robot position by means of an alignment method.

A common problem with all these approaches is the need of an exact a priori knowledge of the landmarks' positions with respect to a metric description of the robot's environment. An attempt to perform landmark-guided navigation in an unknown environment has been done by Trahanias et al. [11] who use landmarks for topological-based navigation. Nevertheless, the patterns of the landmarks they use (e.g., a box or a fire extinguisher fixed to the corridor's walls) need to be previously learnt by the system, and thus also in this case there is an (partial) a priori knowledge on the environment. Moreover, the approach is potentially unstable, because the objects selected as landmarks can change their positions.

In this paper we propose to directly recognize the different kinds of topological elements which characterize the robot's environment. Rather then trying to associate arbitrarily chosen objects or landmarks to specific environments, we propose to directly classify the sub-environments the robot meets during its navigation-discovery by the exploitation of their perspective appearance. We assume that the robot works in an hallway environment, hence the topological elements the system needs to distinguish are: *straight corridor* (the two corridor's walls in the proximity of the robot are not interrupted by corners or crossings: e.g., see Figure 1 (a)), *left* or *right corner* (the robot is next to a wall with the possibility to turn only on the left or only on the right, as in Figure 1 (c)), *end of corridor* (a wall is in front of the robot without corners), *T-junction* (a wall is in front of the robot with two corners) and *crossing* (as in Figure 1 (b)).

The recognition process is based in an incremental reconstruction of the main structural elements of the hallway, as they appear in a perspective view. First, the system performs a standard edge detection and polygonal approximation of the current camera frame. The segments extracted from the polygonal approximation are then used in order to find the z-axis vanishing point with a modified Hough Transform approach [1]. There is a large literature on vanishing point search techniques (see, for instance, [3]). Our aim here is not to propose a general-purpose vanishing point search method but to exploit the assumptions on the robot's environment (a corridor in which main information about the walls can be described using segments) to perform accurate and real-time computations. Once the vanishing point has been found in the image, we use a Hough-like approach to find the edges of the floor. Finally, we analyze the edges of the walls and of the floor in order to understand the topological configuration of the corridor. The computational complexity of each of the previous steps is at most squared with respect to the number of segments and the whole execution time takes only few second's fractions.

2 Efficient Vanishing Point Search

First of all we perform a standard edge detection and thinning process by using the Canny edge detector with Sobel 3×3 masks [7]. Then the edge points are grouped in lines (i.e. continuous sets of points in which each point is adja-

(a) (b) (c)

Fig. 1. Three different types of topological hallway environments: (a) a *straight corridor*, (b) a *crossing* and (c) a *right corner*.

cent to at most two other points) and the lines approximated using a standard *split* polygonal approximation method [7]. The segments so obtained are then used as atomic elements of the subsequent elaborations. This leads to both save computational time (since the number of segments belongs to a lower order of magnitude with respect to the number of edge pixels) and reduce noise from data. In fact, since our scene understanding is based on the analysis of walls and walls' intersections, we do not deal with smoothed objects, thus ignoring human people and any other nonpolygonal object possibly present in the corridor.

The segments are represented by means of the tuple: $< \alpha, \rho, P_1, P_2 >$, where α and ρ are, respectively, the angle of the normal and the distance of the straight line passing through the segments. The line is represented by the equation [7]: $\rho = x \cos\alpha + y \sin\alpha$, where the coordinate reference system is fixed in the bottom left corner of the image I and $0 \leq \alpha < 360$, $0 \leq \rho \leq R$, R being the diagonal of I. Finally, $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are the two endpoints of the segment.

Let S be the set of all the segments. Using the values of the α parameter we can classify each segment as belonging to the set of *vertical* segments $V = \{s \in S : s =< \alpha, \dots >, \alpha \in [th_1, th_2]\}$, *horizontal* segments $H = \{s : s \in S : s =< \alpha, \dots >, \alpha \in [th_3, th_4]\}$ and *oblique* segments $O = \{s : s \in S : s \notin V \wedge s \notin H\}$, where th_1, th_2, th_3 and th_4 are prefixed thresholds (presently we set: $th_1 = -5, th_2 = 5, th_3 = 85, th_4 = 95$). Figure 2 (a) shows the sets V, H and O extracted from the image of Figure 1 (b).

In the next step, we search for the z-axis vanishing point using the elements of O. For each couple of segments $s =< \alpha_s, \rho_s, \dots >$ and $s' =< \alpha_{s'}, \rho_{s'}, \dots >$, $s, s' \in O$, we compute the intersection point $P_{ss'}$ between the two straight lines represented by α_s, ρ_s and $\alpha_{s'}, \rho_{s'}$ respectively. $P_{ss'}$ is used to vote in a Hough accumulator A of the same dimensions of I which is initially set to zero ($A(x, y) = 0, \forall (x, y) \in I$). Given the intersection point $P_{ss'}$, the vote action is: $A(P_{ss'}) := A(P_{ss'}) + |s| + |s'|$, being $|x|$ the length of the segment x.

A common, well-known problem with Hough methods is the need to find a suitable grid size for the accumulator's representation [7]. Too fine a grid can produce vote dispersion because different elements of the same object can vote in different accumulator's cells due to discretization and noise effects. On the other

hand, too coarse a grid can produce merging of votes belonging to different objects in a unique accumulator's cell. We deal with this problem by using a fine representation (each cell of A corresponds to a pixel in I) and then clustering the votes using a technique presented in [1]. The clustering is done as follows.

After that the voting operation has been executed for all pairs of oblique segments, A represents the result of the voting process. Since lines which converge in a real image can spread their vote in different cells of the accumulator because discretization and noise effects, we want to find in A the most "heavy" area, which represents the most likely position of the vanishing point. For this reason we scan A using a squared mask $W_{2l+1\times 2l+1}$. The size of W should be chosen in order to let the window to include the expected area of vote dispersion (we set $l = 3$). When W is centered in p we take into account only the votes in the neighborhood of p contained in the mask W. Let us denote with $W(p)$ the set of all the nonzero cells of A contained in the mask W when it is centered on p. For a given set $W(p)$ we compute the "mass" $M(p)$ of $W(p)$, as the sum of the values of the elements of $W(p)$. The maximum found value $\overline{M} = \max_{p\in I} M(p)$ corresponds to the mass of the most likely area for the vanishing point and it can be computed scanning sequentially each row y of A and using a dynamic programming technique [1]. Assuming (for simplifying reasons) to compute $M(p)$ for only those points far at least l from the boarders of I, for a given point p ($p = (x, y)$) the mass $M(p)$ is given by:

$$M(x,y)=\begin{cases}\sum_{p_1\in W(x,y)} A(p_1), & x = l+1,\\ M(x-1,y)-\sum_{p_1\in W_1(x-1,y)} A(p_1)+\sum_{p_1\in W_{2l+1}(x,y)} A(p_1), & x > l+1\end{cases} \tag{1}$$

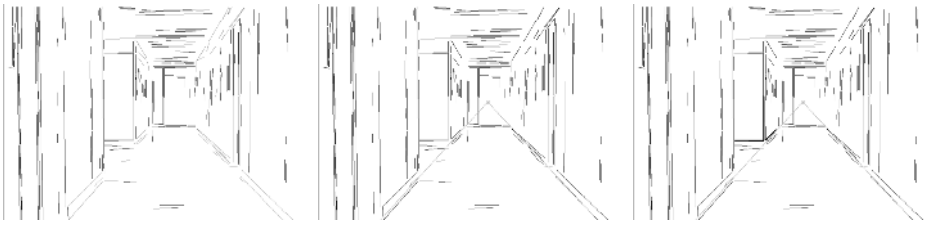
where $W_i(p)$ represents the nonzero elements of the i-th column of the mask $W(p)$. In other words, for a given row y of A, the mass can be computed using the base case of Equation (1) for the first cell and the inductive case for all the others (except the last l cells on the right). If $\overline{p} = \arg\max_{p\in A} M(p)$, then the coordinates (Z) of the z-axis vanishing point are given by the centroid of the elements of $W(\overline{p})$:

$$Z = \frac{\sum_{p_1\in W(\overline{p})} A(p_1)p_1}{\overline{M}}, \tag{2}$$

3 Splitting Lines

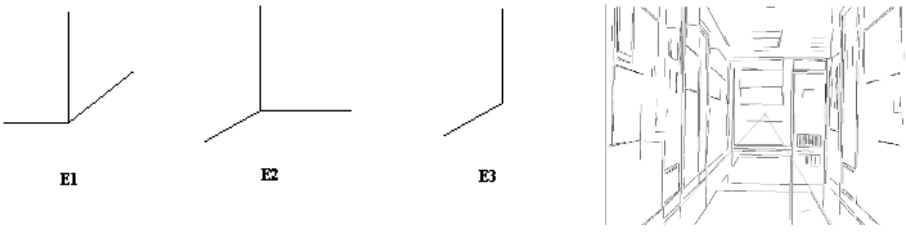
The next step in the perspective structure reconstruction is to find the 2 lines separating the floor from the lateral walls. Unfortunately we cannot simply clustering all the segments converging on the vanishing point Z because these are spread all along the image. Nevertheless, we observe that the (ideal) splitting lines are close to the endpoints of a lot of vertical segments (e.g., the conjunctions of the doors with the floor) and horizontal segments (e.g., the points in which the tails are close to the walls). This information can be exploited in the searching process.

We use for this task a second Hough accumulator A' to represent the parameters α and ρ of all the lines of the image I. A' is initially set to zero.



(a) (b) (c)

Fig. 2. The intermediate results of the recognition process. (a) The sets of segments V, H and O for the image of Figure 1 (b), each set represented with a different graylevel. (b) The same image after the *splitting lines'* extraction. (c) The E_1 type of *topological edges' configuration* recognized on the crossing scene.



(a) (b)

Fig. 3. (a) The three types of topological edges configurations (b) The example of Figure 1 (c) in which the right *splitting line* has been wrongly extracted (see the text).

For each $v \in V$ and each $o \in O$ ($o =< \ldots, P_1^o, P_2^o >$) we verify if v and o are close enough to be considered joined. More in details, for each endpoint $P^v = (x^v, y^v)$ of v we first of all check if $x^v \in [x_1^o, x_2^o]$, being $P_1^o = (x_1^o, y_1^o)$ and $P_2^o = (x_2^o, y_2^o)$. If this is the case, we compute the distance $d(P^v, o)$ of the point P^v from o. If $d(P^v, o)$ is smaller than a given threshold th_5 (we set $th_5 = 10$) then v and o are considered joined and the junction point j_{ov} is computed: $j_{ov} = (x^v, y^v + \frac{d(P^v,o)}{2})$ if v is below o and $j_{ov} = (x^v, y^v - \frac{d(P^v,o)}{2})$ otherwise.

The point j_{ov} is used together with the vanishing point Z in order to compute the parameters (α and ρ) of the line L passing through j_{ov} and Z. L is a candidate *splitting line* and α and ρ are used to increment the accumulator: $A'(\alpha, \rho) := A'(\alpha, \rho) + 1$. The same procedure is repeated also for horizontal segments.

The 2 maximum values in A' after the voting process are retrieved using the same technique described in the case of the vanishing point (Section 2). From now on we will indicate the bottom-left and bottom-right *splitting lines* respectively with L_1 and L_2. Figure 2 (b) shows an example.

4 Topological Edges Configuration

The final step of the proposed incremental scene interpretation is the recognition of the type of topological environment (as they have been defined in Section 1) the robot is looking at by analyzing the information previously extracted. We observe that when a lateral wall is interrupted by another orthogonal corridor (because a *crossing*), the edges of the joining walls form the configuration E_1 shown in Figure 3 (a). Analogously, a *corner* is characterized by the edge configuration E_2 shown in Figure 3 (a). The *end of corridor* is characterized by two configurations of type E_2 (one on the left and the other on the right side of the hallway). A *T-junction* is characterized by two configurations of type E_3. Finally, in a *straight corridor* we cannot observe anyone of the abovementioned configurations (at least not in the proximity of the robot). Thus, the current topological environment can be distinguished by looking for the *topological edges configuration* in the segments' dispositions.

Before to analyze the segments' configurations we first arrange a more suitable representation of the sets of segments S. In the following we show the representation criteria and the edge configurations' search with respect to the splitting line L_1 omitting to show the analogous process for L_2. We use Boolean tables $T_K[x]$ ($0 \leq x \leq x_Z$, being $Z = (x_Z, y_Z)$) initially set to false. $T_K[x] = true$ means that there exists a segment s of type K sufficiently close to L_1 where $P = (x, y)$ is one of the endpoints of s.

For each element $v \in V$ ($v =< \ldots, P_1, P_2 >$) let P ($P = (x, y)$) be the point with the lowest ordinate between P_1 and P_2. We check that P is above L_1, otherwise we discard v: it is noise on the floor. If v is sufficiently large and P is sufficiently close to L_1 we set $T_V[x] := true$. Moreover, for each segment $h \in H$ ($h =< \ldots, P_1, P_2 >$) let $P = (x, y)$ be the point between P_1 and P_2 which is closer to L_1. If h is sufficiently large and P is sufficiently close to L_1 then we set either $T_{H_L}[x] := true$ or $T_{H_R}[x] := true$ depending on whether h is located on the left or on the right side of L_1. Finally, if $o \in O$ ($o =< \alpha, \rho, P_1, P_2 >$) we check that the distance of Z from the line represented by (α, ρ) is no greater than $d = \sqrt{2(\frac{2l+1}{2})^2}$, where $2l + 1$ is the side of the window W used in Section 2 in order to find Z. We discard o if it is not convergent on Z. Furthermore, we discard o if it is not sufficiently large and close to L_1. Without loss of generality, let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ and $x_2 > x_1$. We represent o twofold: both by setting $T_{O_D}[x_1] := true$ and by setting $T_{O_U}[x_2] := true$.

Once built T_V, T_{H_L}, T_{H_R}, T_{O_D} and T_{O_U} we start searching for the three kinds of topological edge configurations E_1, E_2 and E_3. Concerning E_1 the system works as follows. For each x such that $T_{O_D}[x] = true$, we check in a neighborhood of x for the existence of x' and x'' such that $T_V(x') = true$ and $T_{H_L}(x'') = true$. The presence of 3 segments v (represented by $T_V(x') = true$), o ($T_{O_D}[x] = true$) and h ($T_{H_L}(x'') = true$), close to each other and displaced the first vertically, the second (approximately) aligned with L_1 and the third on the left side of L_1 means that we have found a configuration of type E_1. An example of such a triple of segments is shown in Figure 2 (c).

Looking for the E_2 type of configurations follows an analogous process. Let x be such that: $T_{O_U}[x] = true$ *and* $\nexists w : T_{O_U}[w] = true \wedge w > x$, where the second condition is important because we want that x is the closest possible to Z (no other oblique segments can be aligned with L_1 and closer to Z if x represents the point in which the floor meets a frontal wall). Finally, E_2 is recognized if we can find, x' and x'' in a neighborhood of x such that: $T_V(x') = true$ and $T_{H_R}(x'') = true$. E_3 is searched similarly.

As shown in the first part of this section, the combination of the possibly recognized types E_1, E_2 and E_3 allows the system to recognize the topological environment. In the case in which both E_1 and E_2 (or E_3) are recognized, the answer of the system is "crossing" (because E_1 is necessarily closer to the robot). If no type at all is recognized, the system's output is "straight corridor".

5 Efficiency and Experimental Results

For lack of space we have to omit a detailed analysis of the complexity of the whole recognition procedure. However, it is clear that each phase above presented, except the maxima searching in the Hough accumulators, have a computational cost at most squared with respect to the number of segments involved. For example, in Section 3 we look for a candidate splitting line by computing the intersection among all the possible pairs of segments (v, o) and (h, o), for each $v \in V$, $h \in h$ and $o \in O$. If $N = |S|$, this step is $O(N^2)$. The dynamic programming technique used to search for the maximum in the Hough accumulator A (Section 2) allows us to perform the search in $O(Mw)$, being $M = |A| = |I|$ and $w = 2l + 1$. A similar result is obtained for the Hough accumulator A' in Section 3. As we can see, all the involved operations have a very low computational cost and thus can satisfy real-time requisites of a typical robot navigation task.

We have implemented and tested our system with a non-optimized Java code on a Pentium 4, 2.66GHz CPU, with an average execution time of about 0.54 seconds per image. We have used as test data base 87 images randomly taken from 5 different corridors (3 of which are shown in Figure 1). No assumption has been done on the lighting conditions. In the 58.6% of the images the system has correctly recognized *all* the topological environments without any false positive or missing element. In the 78.2% of the images the system has correctly recognized the environment *closest to the camera point of view*. Errors are mainly due to the noise produced by the presence of strong lights on the ceiling or to mirror effects of the walls' edges on the floor (as in Figures 1 (c) and 3 (b)).

6 Conclusions

We have presented an approach to topological navigation for unknown, indoor environments based on the direct recognition of the topological structure of the corridor. By means of an incremental approach we reconstruct the structure

of the walls present in the observed scene. For this reason, the method does not depend on the presence of specific objects (landmarks) whose position in the environment needs to be a priori known. Experimental results have shown real-time performances and good recognition skills.

Acknowledgements

We want to thank the students Elisabetta Cascino, Sergio Garofoli and Valentina Madonna for their precious help in developing and testing the second version of the system's prototype.

References

1. Anelli, M., Micarelli, A., and Sangineto, E. A deformation tolerant version of the generalized Hough transform for image retrieval. In *Fifteenth European Conference on Artificial Intelligence (ECAI 2002), Lyon, France* (2002).
2. Borenstein, J., Everett, H. R., and Feng, L. *Navigating mobile robot: sensors and techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.
3. Cantoni, V., Lombardi, L., Porta, M., and Sicard, N. Vanishing point detection: representation analysis and new approaches. In *11th International Conference on Image Analysis and Processing (ICIAP01)* (2001), pp. 90–94.
4. Courtney, J. W., Magee, M. J., and Aggarwal, J. K. Robot guidanceusing computer vision. *Pattern Recognition 17. No. 6* (1984), 585–592.
5. Dulimarta, H. S., and Jain, A. K. Mobile robot localization in indoor environment. *Pattern Recognition 30. No. 1* (1997), 99–111.
6. Fiala, M. Linear markers for robot navigation with panoramic vision. In *1st Canadian Conference on Computer and Robot Vision (CRV04)* (2004).
7. Forsyth, D. A., and Ponce, J. *Computer Vision: A Modern Approach*. Prentice Hall, 14 August, 2003, ISBN: 0130851981, 2003.
8. Kosaka, A., and Kak, A. C. Fast vision-guided robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP 56. No. 3* (1992), 271–329.
9. Lee, W. H., Roh, K. S., and Kweon, I. S. Self-localization of a mobile robot without camera calibration using projective invariants. *Pattern Recognition Letters 21* (2000), 45–60.
10. Thrun, S. Learning metric-topological maps for inddor mobile robot navigation. *Artificial Intelligence 99* (1998), 21–71.
11. Trahanias, P. E., Velissaris, S., and Orphanoudakis, S. C. Visual recognition of workspace landmarks for topological navigation. *Autonomous Robots 7* (1999), 143–158.

Rectangular Traffic Sign Recognition

Roberto Ballerini[1], Luigi Cinque[2], Luca Lombardi[1], and Roberto Marmo[1]

[1] Dipartimento di Informatica e Sistemistica, University of Pavia, 27100 Pavia, Italy
{luca.lombardi,roberto.marmo}@unipv.it
[2] Dipartimento di Informatica "La Sapienza", University of Roma, Italy
cinque@di.uniroma1.it

Abstract. In this research the problem of the automatic detection and classification of rectangular road sign has been faced. The first step concerns the robust identification of the rectangular sign, through the search of gray level discontinuity on the image and Hough transform. Due to variety of rectangular road signs, we first recognize the guide sign and then we consider advertising the other rectangular signs. The classification is based on analysis of surface color and arrows direction of the sign. We have faced different problems, primarily: shape alterations of the sign owed to the perspective, shades, different light conditions, occlusion. The obtained results show the feasibility of the system.

1 Introduction

In this research the problem of the automatic detection and classification of rectangular road sign (Fig. 1) has been faced. In recent years, much papers on road sign recognition has been proposed with the aim at driving assistance system and autonomous vehicle [2,3,4]. Related to road guide sign set up for guidance at intersection, some approaches can detect a guide sign surrounded by flat intensity sky region in plane image at upper center [1,4]. Moreover, some approaches have been proposed concerning the analysis of the arrows representing directions and the recognition of the character representing destinations and distances, in order to understand information on the sign by assigning the characters to the arrowheads [1,6,7,9]. Our approach can detect sign candidates located at the right hand of the highway even in presence of more complex background. We define five classes of road signs (Fig. 1): advertising, extra-urban, urban, tourist locality, highway. The elements of the last set can strongly differ between them and the others, instead the components of the first four partitions are similar. In such representation, the five classes appear as separated partitions, underlining the difficulty to directly delineate boundaries between them. The Italian Highway Code [5] give information about: 1) shape: rectangular signs are used only with aspect ratio in [0.5, 2.0] and the surface is bounded in 6 m^2; 2) color: colors choice for guide sign: blue for extra-urban, white for urban, brown for tourist locality, green for highway; there are not constraints related to colors for advertising sign, however the Code prohibits in explicit way to use them in combinations that objectively give advertising sign similar to the indication road

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1101–1108, 2005.

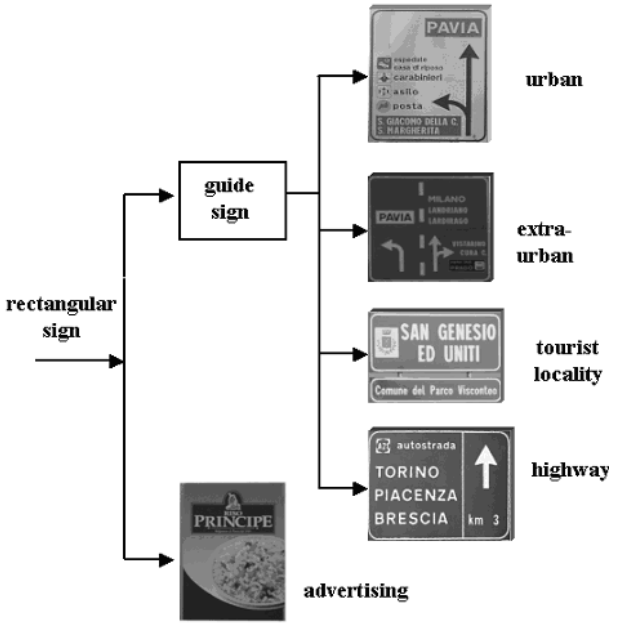


Fig. 1. Classes of rectangular road signs

sign; 3) location: they must be sets to at least height of 1.5 m. from the road level and displaced to at least 3 meters from the right limit of the road.

2 Acquisition System

Our vision system is based on a single camera and bitmap color images of 640x480 pixels, 150 dpi, 24 bits for each pixel. It is important to delimit a small area of interest, minimizing the possibility to get noise from undesired objects. Due to location of the sign, we consider as region of interest the area of 320x480 pixels corresponding to the right hand of the image. Additionally, due to the frequent absence of the road borderlines it is impossible to use the borderlines for delimiting a specific area of interest on the right hand. So, it is necessary to install the camera with an opportune tilt angle q; the value is set to 30 considering the possibility to recognition signs partially hidden by obstacles and the necessity to do not deform the detected shape. Due to horizontal skew between the vision system and the road perpendicular to the sign, the sign appears on a rotated plan (Fig. 2).

The width w' of the sign is less than real w: $w' = w^* xcos(\theta)$. The scale reduction is proportional to the increase of distance among the two extremities of the sign from the camera. The minimum and maximum distances between camera and sign are $d_{\min} = d$ and $d_{\max} = d + \Delta d$ where $\Delta d = w \times \sin\theta$. It is necessary to evaluate the apparent scale reduction of the more distant part

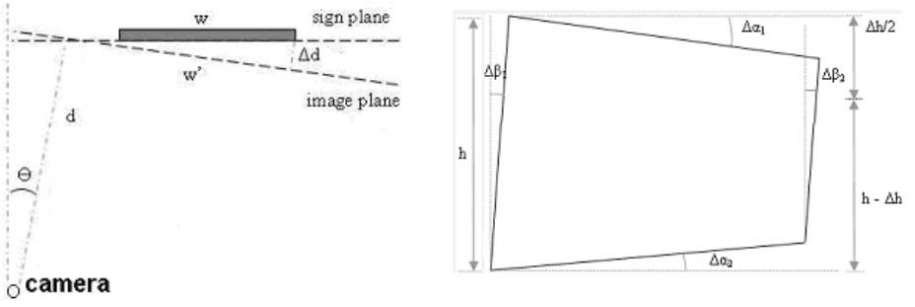


Fig. 2. Left: position of the camera and rectangular sign, right: model of the sign considering rotation and perspective deformation effects

of the sign. Due to perspective alteration, the height of the sign in the left side is reduced by a factor proportional to distance increase from observer in comparison to the left side. The height variation Δh is:

$$\Delta h = \frac{h}{d + \Delta d} \times \Delta d$$

To formalize the model deformation it is necessary to evaluate the angle $\Delta\alpha'$ due to the upper side of the sign and the horizontal line (Fig. 2):

$$\Delta\alpha' = \arcsin\left(\frac{\sin\theta}{2} \times \frac{h}{d + w \times \sin\theta}\right)$$

In order to evaluate the maximum value of the angular inclination of the horizontal sides, it is necessary to maximize $\Delta\alpha'$ in the range for the parameters d, w, h. It is necessary to consider that the signs are not perfectly parallel to the road level, supposing that vision system is parallel to the road and therefore perpendicularly to the sign; we need to introduce Da" that include this rotation. Instead the vertical sides are only subjects to this rotation factor, so $\Delta\alpha = \Delta\alpha' + \Delta\alpha''$ and $\Delta\beta = \Delta\alpha''$ and the deformations for horizontal sides are $\Delta\alpha_1$ and $\Delta\alpha_2$, for vertical sides are $\Delta\beta_1$ and $\Delta\beta_2$.

3 Sign Recognition

We decompose the recognition in three steps:

1. Identification, to verify the presence of a shape interpretable as a sign;
2. Validation, a set of controls in order to verify the real nature of the object;
3. Classification, to distinguish advertising signs from others guide signs.

3.1 Identification

The first step concerns the robust identification of the rectangular signs. Route guide sign assumes a rectangular shape, so we extract the four sides of rectangular shape in order to consider rectangles which matches most likely to the

sign. The intensity of such sign is uniform on its surface and it is different from background. Therefore we analyze the level of contrast and dynamically establish thresholds to identify the four sides applying Canny edge operator, searching gray level discontinuity on the image. In order to consider only the most interesting contours or rather those characterized by more elevated contrast, a threshold is used to consider only a specific percentage p of pixel of the image. Based on the histogram of the module of the gradient, the first index s is chosen according to value of the module that has accumulated relative frequency equal to $1 - p$ so we have:

$$threshold(p) = min_s : \left(\sum_{i=0}^{s} f_r \left(G_m \right) \geq \ 1 - p \right)$$

After the edge detection step, it is necessary to look for line segment to verify the presence of a sign. Our approach is based on Hough transform. We do not use the generalized Hough transform to directly detect the rectangular shape, because the prospective deformation of the shape creates much quantity of possible candidates and, consequently, an high computational effort. For each interesting edge pixel we vote for all the possible straight lines that pass in the edge pixel and that have compatible inclination of the sides in the formalized model of the sign (Fig. 2). In this way it is possible to achieve a precise identification even in presence of noise and in case of partially hidden signs.

At this point the identification looking for n horizontal lines (accepting a maximum error $\Delta\alpha$) and the n vertical lines (accepting a maximum error $\Delta\beta$) which get more vote. The choice of n tightly depends on the complexity of the image: in case of only signs is sufficient $n = 2$ (minimal theoretical value) to exactly detect it is shape; in other situations, characterized by the presence of noise objects (stakes, vehicles, buildings, etc.) good results are obtained only choosing $n > 5$. If this step has not identify an useful line set, the computation conclude that there is no sign in the image. On the contrary it is necessary to establish a criterion to select only the straight lines that effectively correspond to sign edges.

Now it is necessary to intersect every couple formed by an horizontal and a vertical line to find the vertexes of the best approximating quadrilateral. The number of shapes to be examined (N) quickly grows with n. Problem complexity is $O(n^4)$, so it is necessary to use the least value of n that allows good results. All the localized quadrilaterals are now submitted to a sequence of test, in order to identify only the quadrilaters that contain advertising sign.

Eleven tests have been developed, separate in two categories. The first series of test is constituted by nine tests that represent some sufficient conditions for quadrilaterals invalidation. The quadrilaterals are sequentially examined, and if at least a condition is verified the correspondent quadrilateral is eliminated. The second set of tests analyzes the remaining quadrilaterals, through more articulated inspections, attributing a score to each. The phase of identification ends therefore with the selection of the figure that pass all the tests and that has gotten the highest score. If no quadrilateral reaches the end we proceed to re-analyze the image with a recovery strategy.

3.2 Validation

The previous process ends with the identification of the coordinates of the best fitting quadrilateral. In order to exclude the false positive, it is necessary to identify some numerical property. The false shapes in which the system frequently fall are created by buildings and vehicles on the road. A specific analysis of chromatic and intensity characteristics of such objects suggest to study an approach based on depth perception. A sign is a plane nearly perpendicular to the observer situated to a constant distance from it. On the contrary, a false positive is generated by different objects that form a sign shape without this characteristic.

Obviously, a near and well defined object have elevated gradients related of the border. The same objects locate to greater distance loss in definition and will be characterized, therefore, from minor intensity variations. The result is the uniformity of the gradients, that we can formalize as low variance. The sign image is characterized by a greater dispersion of values in the whole range.

If the sign search is failed with a specific set of parameters, we can suppose that if we repeat the sign search with a new set of parameter value, best results can be gotten. If the image has low contrast it is probably that in the previous gradient threshold calculation we do not consider all the interesting pixels. Consequently, it is necessary to repeat the line segment detection with new thresholds. In order to reduce the effect of the noise, we control the votes received by the selected straight lines. Two different method are used to determine if such lines have been produced by single object points, or if they have been produced by a group of objects:

- Backward lines checking. First method: to valuate in the Hough space the continuity of the points that give contribute to accepted lines (digitalization errors make borders of the objects frequently fragmented, making to appear a continuous line as discontinuous). Second method: to verify in any accepted line the presence of a good point concentration.
- Recovery strategy. In this way we realize a validation step of the segments detected with modified parameters. With these straight lines the survey is repeated and in case of positive result the search finishes. On contrary case this correction step is repeated, reducing at every step the parameters that control it, that are threshold on the intensity gradient module and threshold on the density of votes along the straight lines. The performances really improve, making the system more robust in situations characterized by non optimal input image.

The test set includes 1100 images: 900 containing a rectangular sign, 200 without rectangular signs. This approach can recognize the presence of a rectangular sign with 99% accuracy. The obtained results are compared with the position of the rectangular sign obtained by a human user. If the background colors are very similar to colors of advertising sign, it is not possible to detect the rectangular shape. The system is able to repeatedly detect the same sign in four consecutive images.

Fig. 3. Model of arrow direction (left): h is the height, w is the width, a is the direction angle. White line corresponds to the analyzed column (center) and the related analyzed rows (right) in extra-urban sign.

3.3 Classification

Due to variety of rectangular road signs, we first identify guide road sign and then we consider all others as advertising sign. So, it is necessary to create a set of features that characterizes the five classes (Fig. 1). Analyzing the Italian Code [5], we can understand that the first information to be examine is the percentage of the colors that are painted on the sign surface. The Code fixes the colors that must be use distinguishing them by the location and the functionality which they are created in the sign. Using the RGB color model, the frequent color is the color that covers the 50% of the sign surface (for example white for urban sign), the information color is used to show information on the sign (for example black for urban indication sign). The thresholds are empirically defined by statistic measurements on the entire set of images evaluating lighting conditions as reflections and shadows. At this point the algorithm search a characteristic element: the arrows direction. The purpose of these symbols is to facilitate the car driver in the choice of the way. To avoid confusion of the driver the Code explicitly prohibit to use the arrows direction in the advertising road sign. Moreover, the Code establishes: a specific shape, the ratio between height and width is h/w=10/11, arrowhead direction angle a in [0, 45, 90, 135, 180] (Fig. 3).

Approaches based on contour description [7], template matching and neural networks [9], morphological operations [4] that require a considerable computational effort and they are sensitive to noise. We consider the white arrow due to highway and extra-urban sign and the black arrow due to urban sign. Our approach is based on the following steps:

1. to analyze each column of the matrix representing the image (Fig. 3);
2. if there is more than 70% of white or black pixels it is necessary to analyze the rows in the matrix composed by pixels near to the top of the white or black pixel (Fig. 3);
3. if the cluster has the shape of a triangle then there is an arrowhead.

The cluster has a triangular shape if these conditions are both satisfied:

1. a correspond to number of white or black pixel in row y, b correspond to number of white or black pixel in row $y+1$, so $a = b$ or $a+1 = b$ in progress for each row starting from the top row y^*;

2. the ratio between number of rows and number of columns in [0.8, 1.0];
3. the ratio between height and width of the cluster of pixel is roughly 10/11.

In case of negative result, the steps are repeated using a new value of angle a, in order to rotate the candidate sign. If there is a compatible arrow the classification step finishes establishing the class of the sign. Our approach is very fast and robust: using a test set of 300 guide signs image the accuracy of correct recognition is 100%. Finally, the algorithm verify the presence of typical rectangular frame that characterizes the guide sign using Hough transform to detect the four straight lines only at the boundary of the sign.

3.4 Performance Evaluation

The test set is formed by 900 images: 500 containing advertising, 400 containing guide sign, that is, 100 images for each guide sign class. This approach can recognize the presence of a rectangular sign with 96.6% mean accuracy, the accuracy for classification of specific guide sign is 96.5%. The confusion matrix (Tab. 1) allows to summarize the classification results on the test set.

In case of erroneous classification of the guide sign, the class chosen is the advertising one, it may be seen that it does never occur that an image of an urban sign is classified as tourist or highway. Two advertising signs are misclassified as urban due to high quantity of white and black colours, one advertising sign is misclassified as tourist due to high quantity of brown and white colour. Four tourist signs are misclassified as advertising due to presence of a photo of the

Table 1. Confusion matrix: c_{ij} coefficient at row i and column j represents the percentage of sign of class i identified as sign of class J

predicted classification	real classification				
	advertising	extraurban	urban	tourist	highway
advertising	97	0	2	1	0
hline extraurban	3	97	0	0	0
hline urban	4	0	96	0	0
hline tourist	4	0	0	96	0
hline highway	3	0	0	0	97

Fig. 4. Classification results indicated by the black frame: urban sign (left) and advertising (center), an advertising sign partially hidden by a pole (right).

locality. The developed system allows good results in many situations (Fig. 4). In some images, due to presence of noise or partially hidden signs only a part of surface sign has been identified. However, this part is sufficient to complete in a corrected way the classification (Fig. 4). Some problems arise when the sign is too far, or when the sign has very low contrast considering the surrounding environment.

4 Conclusion

In this paper we have proposed an image analysis that allowed to recognize the rectangular road sign in colour images distinguishing them between advertising and guide sign. This technique predicts the advertising sign with 97% accuracy and the specific type of guide sign with 96.5%, even in presence of perspective deformation, different conditions of brightness, partially hidden signs. A robust method for arrow direction recognition is proposed.

Experimental results show the effectiveness of the approach. This percentage is satisfactory for this preliminary application of the proposed methodology. The codes have been written in C language as part of a package, which can be used and extended for future applications. Future works include integration with a pre-attentive optical flow created by a camera mounted on a moving car.

References

1. Azami, S., Katahara, S., Aoki, M.: Route guidance sign identification using 2-d structural description. Proceedings IEEE Int. Conf. Intelligent Transportation Systems (1996) 153-158
2. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic detection and recognition of signs from natural scenes. IEEE Trans. Image Processing 13 (2004) 87-99
3. De La Escalera, A., Armingol, J. M., Pastor, J. M., Rodriguez, F. J.: Visual sign information extraction and identification by deformable models for intelligent vehicles. IEEE Trans. Intelligent Transportation Systems 5 (2004) 57-68
4. Franke, U., Gavrila, D., Gorzig, S., Lindner, F., Paetzold, F., Wohler, C.: Autonomous driving approaches downtown. IEEE Trans. Intelligent Transportation Systems 13 (1999) 40-48
5. Italian Highway Code. http://www.dirittoegiustiziaonline.it/infortunistica/codst2.htm
6. Kato, T., Kobayasi, A., Hase, H., Yoneda, M.: An experimental consideration for road guide sign understanding. Proc. IEEE Int. Conf. Intelligent Transportation Systems (2002) 268-273
7. Lee, J., Jo, K.: Traffic Sign Recognition by division of Characters and Symbols Regions. Proc. 7th Korea-Russia International Symposium (2003) 324-328
8. Miura, J., Kanda, T., Shirai, Y.: An active vision system for real-time traffic sign recognition. Proc. IEEE Int. Conf. Intelligent Transportation Systems (2000) 52-57
9. Priese, L., Lakmann, R., Rehrmann, V.: Ideogram Identification in a Realtime Traffic Sign Recognition System. Proc. IEEE Int. Conf. Intelligent Transportation Systems (1995) 310-314
10. Salgian, G., Ballard, D. H.: Visual routines for autonomous driving. Proc. 6th Int. Conf. Computer Vision (1998) 876-882

Study of the Navigation Parameters in Appearance-Based Navigation of a Mobile Robot

Luis Payá, Oscar Reinoso, Arturo Gil, Nicolás García, and Maria Asunción Vicente

Departamento de Ingeniería de Sistemas Industriales,
Miguel Hernández University, Avda. De la Universidad s/n,
03202 Elche (Alicante), Spain
{lpaya, o.reinoso, arturo.gil, nicolas.garcia, suni}@umh.es
http://lorca.umh.es/isa/es

Abstract. Recently, appearance-based approaches have attracted the interests of computer vision researchers. Based on this idea, an appearance-based navigation method using the View-Sequenced Route-Representation model is proposed. A couple of parallel cameras is used to take low-resolution fontal images along the route to follow. Then, zero mean cross correlation is used as image comparison criterion to perform auto-location and control of the robot using only visual information. Besides, a sensibility analysis of the navigation parameters has been carried out to try to optimize the accuracy and the speed in route following. An exhaustive number of experiments using a 4-wheel drive robot with synchronous drive kinematics have been carried out.

1 Introduction

There are three families of techniques to carry out the navigation of a mobile robot in indoor environments and without a previous map: model-based techniques [3], [4], appearance-based techniques [8] and optical flow-based techniques [11]. Optical-flow based techniques take into account the relative movement of the scene elements to calculate the control law for the robot. Model-based techniques use natural or artificial landmarks from the scene as references to guide the robot through the desired route. The recognition of patterns is achieved comparing features of the input image with features that have been previously stored. These techniques suppose high complexity due to the difficulty in the features extraction and the comparison of patterns in realistic and changing environments. At last, appearance-based techniques use the appearance of the whole scene. This approach consist on two phases, the learning one, where the robot stores general visual information from several points of view along the route to follow, and the autonomous navigation, where the robot follows the route comparing the current visual information with the stored one. The main disadvantage of these techniques is that they require huge amounts of memory and they suppose high computational cost to model the route and make the necessary comparisons during the autonomous navigation.

In structured environments, auto-location can be performed distinguishing landmarks inside the scene, so we can use model-based techniques. As an example, [12] presents the application of the visual servoing approach to a mobile robot. The robot

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1109 – 1116, 2005.

is controlled according to the position of some features in an image and obstacle detection is outperformed throughout a combination of vision and ultrasonic sensors. Nevertheless, precision in the navigation will depend on the ability to recognize this set of features, what may become very difficult in most situations. Besides, additional problems may arise in non-structured environments, where artificial landmarks cannot be allocated or there are no natural landmarks that could be segmented with precision. In this case, using the properties of global appearance of the images could result more adequate. Also, due to the constant improvement of computer technology, appearance-based approaches have become a feasible technique despite its computational cost. Anyway, the key points to carry out a successful navigation with these methods reside on the quantity of information to store and in how to make the comparison between the current view and the stored information to reduce the computing time.

Researchers have proposed several methods to outperform auto-location and navigation based on global appearance. Matsumoto et al. [7], [8] addressed the VSRR method, consisting on the direct comparison of low-resolution images. Jones et al. [2] proposed a method based on the same concept but using a couple of cameras and odometer information to carry out navigation. Other approach makes use of the color histogram to perform auto-location. The problem is that the histogram does not contain any spatial information, so it is necessary to extract other features, as texture, density of edges, etc., as in [13], [15]. Regini et al. [10] proposed a method that calculates spatial relationships between the regions of color. Also, the complexity of the problem can be reduced working in the PCA space as in [1], [6], or trying to extract the regions of the image that contain the most significant information as in [14].

The approach addressed in this paper is based on the VSRR model with a couple of cameras, but with the objective of following pre-recorded routes using only visual information, with no odometer information. New control laws for the linear and steering velocities are proposed and it is also carried out a study of the parameters that affect the navigation to optimize the accuracy and speed in the route following.

The paper is organized as follows. Section 2 describes the key points of the developed application. Section 3 studies several parameters that have influence over the robot navigation and their optimization. Finally, section 4 exposes the conclusions and the future work that can be developed to improve navigation.

2 Visual Navigation Using Appearance Features

To develop the application, the B21r mobile robot has been used. This robot has 4-wheel drive with synchronous drive kinematics. It means that the four wheels can spin around the vertical axis which passes through their centers, remaining parallel all time. The separation between the driving and the steering systems makes possible to control independently the linear velocity v (robot advance) and the steering velocity ω (robot direction). These two variables will be used to control the robot movement.

On the top of the robot, there is a pan-tilt unit with a couple of Sony XC999 cameras with their optical axis aligned. The maximum resolution of these cameras is 640x480 pixels but an image server that provides images of less resolution has been added. The simultaneous use of two cameras will make our method more robust.

Fig. 1. Material used to develop the application. B21r Robot and Sony XC999 cameras.

The proposed approach is based on the VSRR model [4]. This model solves the problem of the huge memories using low resolution images to model the route.

With the purpose of following pre-recorded routes two phases need to be accomplished: a learning phase in which some visual information about the route is stored and an autonomous navigation phase, in which the current position of the robot must be estimated to drive it through the learnt route. To avoid the vulnerability to environment changes such as unknown obstacles, walking persons and cast shadows, a new phase is being implemented, to detect obstacles using just the two cameras to sense the environment.

2.1 Learning Phase

Firstly, the route is decomposed in straight segments, and the robot is guided in a teleoperated way through these segments, taking images simultaneously with both cameras in several points of the route. In these points, it is also stored, in a qualitative way, the next action the robot must take to follow the route. This action is stored as 'f' when the robot has to drive forward, 'r' when the robot has to stop and turn right, 'l' when it has to turn left or 's' when the end of the route has been reached and the robot must stop. All this information (images and actions) is stored in text files. In the turning points, just two pairs of views are stored, one at the beginning and one at the end of the turn.

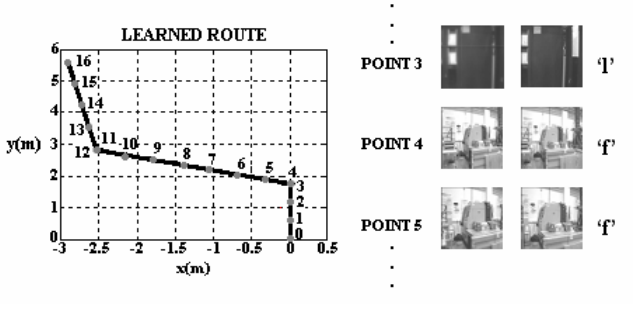


Fig. 2. Form of the database created during the learning phase

2.2 Autonomous Navigation

During the autonomous navigation, the robot is located in a point near the learnt route. Then, it has to recognize which of the stored positions is the nearest to the

current and drive to tend to the route, following it till the end. To do this, two processes, that are executed successively, have been implemented: auto-location and autonomous navigation. The robot will drive forward in the straight segments with a correcting turning speed to tend to the route, and it will have a pure rotation movement in turning points.

Auto-location. To carry out auto-location, the current entire images are compared with all those previously stored in the database (the current left image with all stored left images, and the same with the right image). The adopted comparison criterion is the zero mean cross correlation:

$$\gamma_i = \frac{\sum_{x,y}\left[I(x,y)-\bar{I}\right]\cdot\left[A(x,y)-\bar{A}\right]}{E_I \cdot E_A}. \qquad E_I = \sqrt{\sum_{x,y}\left[I(x,y)-\bar{I}\right]^2} \quad E_A = \sqrt{\sum_{x,y}\left[A(x,y)-\bar{A}\right]^2}. \tag{1}$$

I and A are the two images to compare, whose average values are $\bar{I}, \bar{A}$ and whose energies are E_I, E_A. Although this is a computationally demanding way of doing image matching, it has been chosen because of its insensibility to the scene illumination and input noise [5], comparing to other criteria such as the direct difference of the images. The average and the energy of the stored images can be calculated in an off-line process, what can save computational cost during navigation. As in each point we have two images, to obtain a general data of comparison we use the arithmetic average of the correlations: $\gamma_{av} = \left(\gamma_{left} + \gamma_{right}\right)/2$.

At the beginning of the navigation, the current images need to be compared with all the stored ones, recognizing the nearest position by means of the average correlation. In the next iterations, it is necessary to compare the current image only with the previously matched and the following one, because navigation is continuous and robot has to pass through all pre-recorded points successively. Then, once the robot has started navigation, the time of processing is independent of the database size, and so, of the length of the route to be followed. Besides, we can take points frequently during learning step to get more accuracy without increasing computational cost.

Control. In this task, we have to correct the robot steering to make it tend to the route and follow it to the end. The behavior of the robot depends on the action that the currently matched images have got associated in the database.

If the associated action is 'f' (go forward), we have to correct the small fluctuations that the robot position may suffer respect the route to follow. To do this, the current images are compared continuously with the previously matched and the next ones (Fig. 3: Auto-location). Once we have a new matching, we take a sub-window in the left matched image and track it on the right matched image, identifying the sub-window of the right image that better correlates with the left one (Fig. 3: Control, step 1. This can be done off-line). As the optical axes are parallel, the vertical offset of the right sub-window respect the left one must be zero. Then, the same process is carried out between the left matched image sub-window and the left current image (Fig. 3: Control, step 2), and finally, between the right images (Fig 3: Control, step 3). The horizontal offsets x_1 (between the left sub-windows) and x_2 (between the right

sub-windows) allow calculating the necessary steering velocity to tend to the route. The linear velocity will be proportional to the average correlation, what means that when we are far from the route, the linear velocity will be low to allow the robot correct its trajectory, but when we are following the route quite good, the robot goes quicker. Then, the proposed control law is:

$$\omega = k_1 \cdot x_1 + k_2 \cdot x_2 .$$
$$v = k_3 \cdot \gamma_{av} . \tag{2}$$

Being k_i three constants whose value will be studied in the next section.

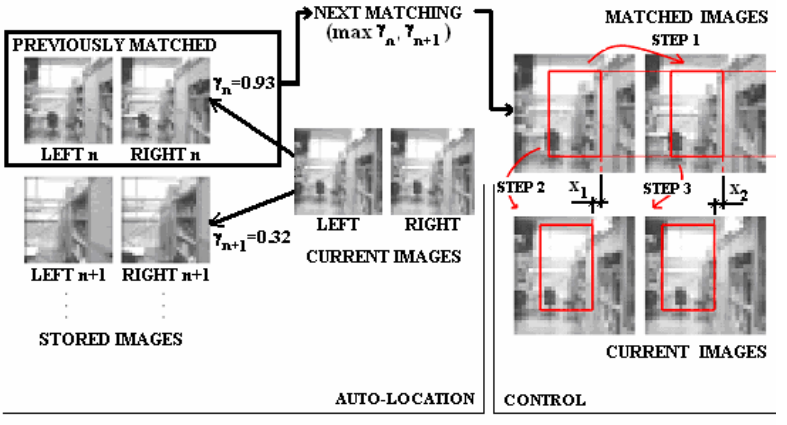


Fig. 3. Tasks performed during autonomous navigation when the action is 'f'. First one, the robot makes auto-location, comparing current images with the previously matched and the next ones. Once we have a match we calculate the linear and steering speeds based on the global correlation and the horizontal displacement of a template.

If the associated action is 'r' or 'l' (turning right or left), the robot keeps going forward according to equation (2), until the average correlation has arrived to a maximum and decreases a 10% respect to this maximum. This is to avoid the fact that the robot begins turning too soon. In this point, the robot stops and begins turning, comparing the current images with the following stored. The steering velocity is inverse proportional to global correlation. When it begins turning, correlation is low, so the robot turns quickly, and as we go closer the final position, correlation increases, so the speed decreases. This supposes a compromise between speed and accuracy.

$$w = k_4 / \gamma_{av} .$$
$$v = 0. \tag{3}$$

3 Analysis of the Navigation Parameters

In this section we study the influence of the parameters previously presented in the behavior of the robot. To carry out this analysis the effects of the linear speed, shape and size of the comparison template and steering speed have been studied separately.

In all the shown experiments, the learnt route, is composed by 16 couples of simultaneous images along a total length of about 8 meters, with two turning points, one to the left and the other one to the right. A resolution of 32x32 pixels has been chosen to capture the images. This allows a top speed of about 1 m/s. Lower resolutions do not provide information enough so the robot gets lost often. Higher resolutions suppose an important increment of calculations at each iteration so the robot speed has to decrease in the same proportion. As an example, a 64x64 resolution allows a top speed of only about 0.3 m/s.

3.1 Influence of the Advance Speed

During the navigation, the advance speed of the robot is a linear function of the global correlation, according to (2). So the advance speed can be modified simply changing the value of the constant k_3. Fig. 4 shows the value of the average correlation during all navigation (average for all the current images during the navigation) for different values of k_3, This value is taken as a measure of the following accuracy, so the accuracy in the following of the learnt route decreases as the speed is higher. Then, we will have to arrive to an agreement between speed and accuracy, depending on the application. For values higher than 0.8, the navigation is not possible. In this case, the robot runs a too long distance with the same control action so, when it is updated, the robot is already lost.

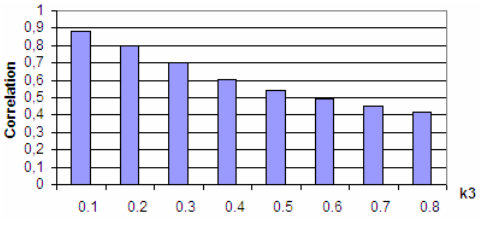


Fig. 4. Results of the navigation accuracy for different linear speeds

3.2 Shape and Size of the Comparison Template

The control during navigation is based on the continuous comparison of a template that is taken on the matched stored images over the current images. So, shape and size of the template result to be parameters of great importance, in part, due to the fact that the computational cost depends directly on the size of this template (for example, using a 24x10 template instead of 16x16 saves a 33% of operations). Exhaustive experimentation makes us arrive to the following conclusions.

The optimal size is the one that has about the half of rows and columns of the complete images. For bigger sizes, the error in the route following increases quickly. In this case, if the deviation respect the learned route is high, part of the template will not appear in the current image. As shown in fig. 5, navigation may become impossible in spite of the fact that the time per iteration is quite low. At last, templates with higher number of rows than columns present worse results.

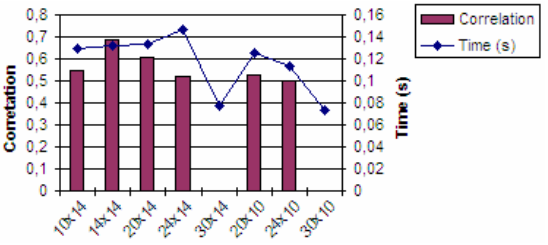


Fig. 5. Results of the correlation and time per iteration for different sizes and shapes of the comparison template. A zero correlation means that navigation has not been possible.

3.3 Influence of the Steering Velocity

The corrections on the robot trajectory so that it follows the learnt route are calculated as a steering velocity that depends on the horizontal offsets between the current images and the sub-windows on the stored ones. Taking into account this fact, the effect of varying the turning speed can be appreciated changing the values of the constants k_1 and k_2. The experiments show that the navigation is very little sensitive to the variation of these parameters. When they take values in [0.001, 0.040], the changes of these parameters do not affect the global error in a noticeable way.

4 Conclusions and Future Work

In this work, new control laws for appearance based navigation with two cameras have been proposed and tested, carrying out a sensibility analysis of the navigation parameters to try to optimize the accuracy and the speed in the task of following prerecorded routes. With these laws, the robot is able to find itself and follow the route in a band of about 2 meters around it. It can be done although the scene suffers changes (illumination, position of some objects…). There are some future works that could improve our navigation system, such as the automation of the learning phase and the implementation of continuous navigation. An automated learning phase would imply the robot should just be driven along the desired route, but the image acquisition would be carried out automatically, when the correlation of the current image respect to the last one stored would go down a threshold. On the other hand, the continuous navigation would allow navigation in any route, without decomposing it in straight segments, so, no information about the action that the robot takes during learning must be stored. That would suppose an important change in the control law.

Acknowledgements

This work has been supported by Ministerio de Educación y Ciencia through project DPI2004-07433-C02-01 ‘Herramientas de Teleoperación Colaborativa. Aplicación al Control Cooperativo de Robots’.

References

1. Artac, M., Jogan, M., Leonardis, A.: Mobile robot localization using an incremental eigenspace model. Proceedings of the IEEE International Conference on Robotics & Automation (2002) 1025-1030
2. Jones, S.D., Andersen, C., Crowley, J.L.: Appearance based processes for visual navigation. Proceedings of the IEEE International Conference on Intelligent Robots and Systems (1997) 551-557
3. Kosaka, A., Kak, A.C., Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. Computer Vision, Graphics and Image Processing: Image Understanding. Vol 56, no3 (1992) 271-329
4. Lebegue, X., Aggarwal, J.K.: Significant line segments for an indoor mobile robot. IEEE Transactions on Robotics and Automation. Vol. 9, no 6 (1993) 801-815
5. Lewis, J.P.: Fast normalized cross-correlation. Expanded version of paper from Vision Interface (1995) 120-123
6. Maeda, S., Kuno, Y., Shirai, Y.: Active navigation vision based on eigenspace analysis. Proceedings IEEE International Conference on Intelligent Robots and Systems. Vol 2 (1997) 1018-1023
7. Matsumoto, Y., Ikeda, K., Inaba, M., Inoue H.: Visual navigation using omnidirectional view sequence. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (1999) 317-322
8. Matsumoto, Y., Inaba, M., Inoue, H.: Visual navigation using view-sequenced route representation. Proceedings of IEEE International conference on Robotics and Automation. Vol 1 (1996) 83-88
9. Ohno, T., Ohya, A., Yuta, S.: Autonomous navigation for mobile robots referring pre-recorded image sequence. Proceedings IEEE International Conference on Intelligent Robots and Systems. Vol 2 (1996) 672-679
10. Regini, L., Tascini, G., Zingaretti, P.: Appearance-based robot navigation. Proceedings of the Workshop su agenti robotici, Associazione Italiana per l'Intelligenze Artificiale, VIII Convegno. (2002)
11. Santos-Victor, J., Sandini, G., Curotto, F., Garibaldi, S.: Divergent stereo for robot navigation: Learning from bees. Proceedings IEEE International conference con Computer Vision and Pattern Recognition (1993) 434-439
12. Swain-Oropeza, R., Devy, M., Cadenat, V.: Controlling the execution of a visual servoing task. Journal of Intelligent and Robotic Systems. Vol 25, No 4 (1999) 357-369
13. Ulrich, I., Nourbakhsh, I.: Appearance-based place recognition for topological localization. Proceedings IEEE International Conference on Robotics and Automation (2000) 1023-1029
14. Winters, N., Santos-Victor, J.: Information Sampling for optimal image data selection. Proceed of the 9th International Symposium on Intelligent Robotics Systems (2001) 249-257
15. Zhou, C., Wei, T., Tan, T.: Mobile robot self-localization based on global visual appearance features. Proceedings of the 2003 IEEE International Conference on Robotics & Automation (2003) 1271-1276

SVM Based Regression Schemes for Instruments Fault Accommodation in Automotive Systems

Domenico Capriglione, Claudio Marrocco, Mario Molinara,
and Francesco Tortorella

Dipartimento di Automazione, Elettromagnetismo,
Ingegneria dell'Informazione e Matematica Industriale
Università degli Studi di Cassino
03043 Cassino (FR), Italy
{capriglione, c.marrocco, m.molinara, tortorella}@unicas.it

Abstract. The paper deals with the use of Support Vector Machines (SVMs) and performance comparisons with Artificial Neural Networks (ANNs) in software-based Instrument Fault Accommodation schemes. As an example, a real case study on an automotive systems is presented. The ANNs and SVMs regression capability are employed to accommodate faults that could occur on main sensors involved in the operating engine. The obtained results prove the good behaviour of both tools and similar performances have been achieved in terms of accuracy.

1 Introduction

Modern automatic measurement systems often require a suitable on-line instrument diagnosis. In fact, nowadays, the most common process control systems base their working on some quantity measurements performed by a suitable set of sensors. Thus, instrument fault detection and isolation (IFDI) schemes are required in many applications where the correct working of the system plays a fundamental role to grant both user/operator safety and efficient process operating [1]-[2].

Automotive systems belong to these contexts, since in the last decade cars, busses and trucks have been equipped with a lot of sensor-based electronic systems devoted to grant the passenger safety and comfort (Anti-lock braking system, Anti-spin regulation, Electronic stability program, Airbag, air conditioning, and so on), as well as to control fuel injection, ignition and pollution emissions of the engines. The correct functioning of such systems strongly depends on the accuracy of the collected measurements and on both the reliability and the status (faulty-free or faulty) of the corresponding sensors. Then, in the last years, as in other critical frameworks (aerospace and nuclear plants), automotive environment has widely growing employing instrument diagnostic techniques with the aim of improving both the vehicle run safety and the engine functioning.

The basic principles of diagnostic are all based on either physical or analytical redundancy, though physical-based redundancy solutions require a double or triple number of identical sensors for each measurand. A more suitable approach is based on model redundancy, thus reducing costs and dimension [3].

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1117 – 1124, 2005.

More in detail, an IFDI procedure consists of two main sections: *i) fault detection,* to point out a fault occurred at least on one sensor, *ii) fault isolation*, to locate the previously detected fault, thus identifying the faulty sensor that will give wrong information.

In addition to these instrument diagnostic capabilities, an important improvement could be obtained equipping the system with sensor *fault accommodation* capability (IFDIA scheme). In this way, the occurred sensor fault will be overcome, thus avoiding the malfunctioning or the stop of the vehicle. As a consequence, the sensor fault accommodation activity is a very attractive technique that at the present is rarely implemented on the modern vehicles. Moreover, the few actual implementations are often based on some very rough models not able to approximate the faulty sensor output with suitable accuracy. On the other hand, artificial intelligence techniques have been developed for automotive systems and good results were achieved for both instrument diagnosis and instrument fault accommodation. In [4]-[5], IFDIA solutions employing artificial neural networks (ANNs) are proposed and tested. They provide a suitable scheme characterized by very interesting features in terms of promptness, sensitivity and accuracy. In particular, they are able to detect, isolate and accommodate sensor faults that could occur on the main sensors involved in a FIAT 1,242 litres Spark Ignition Engine. Starting from such results, and focusing the attention on the *accommodation section*, in this paper, to overcome the sensor faults, we test the capability of the Support Vector Machine (SVM) [6]. This is a recently introduced technique based on Structural Risk Minimization which can be employed both on pattern classification [7] and on regression [8]. After a description of the system under analysis (section 2), section 3 provides a brief introduction to the SVM for regression (SVR). Section 4 illustrates the system architecture, section 5 describes the experimental test bed and finally, section 6 provides a synthesis of the experimental results obtained comparing ANN and SVR.

2 The System Under Analysis

The two different tools for the fault accommodation (ANN and SVR) have been developed and tested on a FIAT 1,242 litres Spark Ignition Engine, four cylinders. The electronic control is based on a speed-density multi-point injection system by Magneti Marelli. Among all, the manifold pressure, the crankshaft speed and throttle valve angle position sensors provide the basic information concerning engine states (i.e. engine load, speed). Thus, the main control actions (i.e. injection time, spark advance) are based on computations performed on the data measured with these sensors. The air induction process is governed by both stationary and dynamic complex thermo-fluid-dynamic phenomena that furnish some close links between the pressure inside the intake manifold, the throttle valve opening and the crankshaft speed [9]. According to these considerations it emerges clear that throttle position (*thr*), inlet manifold pressure (p_{man}) and crankshaft speed (*rpm*) sensors assume a relevant importance among all sensors available on the engine. As a consequence, in this framework, the accommodation of faults that could occur on these sensors is very suitable. Sets of experimental data measured on an engine test bench were available while the standard New European Driving Cycle (ECE+EUDC) was running. In

particular, eight sets of about 1100 real samples were acquired in different fault-free operating conditions and spanned in all the nominal range. These samples were used to develop and test both the ANN-based and SVR-based procedures able to accommodate the faults occurring on sensors that measure the three mentioned quantities.

3 Support Vector Machine for Regression

In this section we present a brief introduction to SVR. Suppose the training data: $X=\{(x_1,y_1),\ldots,(x_l,y_l)\}\subset \chi$ x R where $x_i\in R^d$ is the input pattern, $y_i\in R$ is the target value and l is the total number of training samples, randomly and independently generated from an unknown function. In ε-SV regression, the goal is to find a function $f(x)$ that has at most ε deviation from the actually obtained targets y_i. The general form of the prediction function is:

$$f(x)=\langle w,\phi(x)\rangle+b \tag{1}$$

where $\langle\cdot,\cdot\rangle$ is dot product in Ω, i.e. a feature space with a dimension generally different from χ such that $\phi:\chi\rightarrow\Omega$ and $b\in R$. Our goal now is to determine w and b starting from training data, through the minimization of a functional risk [6]:

$$R[f]=\int c(x,y,f(x))dp(x,y) \tag{2}$$

based on the empirical data X. $(c(x, y, f(x))$ is a cost function that determines how we will penalize estimation errors). Since $dp(x,y)$ is unknown it is possible to use X only in order to estimate a function f that minimizes $R[f]$. Formally, we have a convex optimization problem:

$$\text{minimize } \frac{1}{2}\|w\|^2 \text{ subject to} \begin{cases} y_i-\langle w,\phi(x_i)\rangle-b\le\varepsilon \\ \langle w,\phi(x_i)\rangle+b-y_i\le\varepsilon \end{cases} \tag{3}$$

sometimes such a function does not exist or we can tolerate some errors. In the latter case one can introduce slack variables ξ, ξ^* and the problem formulation becomes:

$$\text{minimize } \frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}(\xi+\xi^*) \text{ subject to } \begin{cases} y_i-(\langle w,\phi(x_i)\rangle+b)\le\varepsilon+\xi^* \\ -y_i+(\langle w,\phi(x_i)\rangle+b)\le\varepsilon+\xi \\ \xi,\xi^*\ge 0 \end{cases} \tag{4}$$

where $i=[1,\ldots,l]$, C is the trade-off between the flatness of f and the amount of point that are tolerated out of the insensitive tube.

The optimization problem above can be solved more easily in its dual formulation. It is possible to construct a Lagrange function from both the objective function (the so called primal objective function) and the corresponding constraints, by introducing a dual set of variables. After some reduction we obtain the dual optimization problem:

$$\text{Maximize} \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)k(x_i,x_j) \\ -\varepsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{l}y_i(\alpha_i+\alpha_i^*) \end{cases} \text{ subject to } \begin{cases} \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0 \\ \alpha_i,\alpha_i^* \in [0,C] \end{cases} \tag{5}$$

The prediction function (1) becomes [11]:

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)k(x_i,x_j)+b \tag{6}$$

The Mercer's theorem [10] ensures the availability of such functions $k(x_i, x_j)$, called kernels; some kernels frequently used in SVM applications (both for classification and for regression) are: *polynomial* $k(x_i, x_j)=(< x_i, x_j>+c)^d$, *Gaussian (RBF)* $k(x_i, x_j)=exp(-(1/\delta^2)(x_i- x_j)^2)$, *sigmoid* $k(x_i, x_j)=tanh(\beta < x_i, x_j> + \gamma)$ where *c, d,* δ, β and γ are kernel parameters.

The computation of *b* can be done exploiting the so called Karush-Kuhn-Tucker (*KKT*) conditions according to which the product between dual variables and constraints has to vanish at the optimal solution point. From *b* computation it follows that only for $|f(x_i) -y_i| \geq \varepsilon$ the Lagrange multipliers may be non zero, or in other words, for all samples inside the ε-tube the α_i, α_j^* vanish. The examples that come with non vanishing coefficients are called *Support Vectors*.

4 The System Architecture

On the basis of previous experiences in the analyzed field ([4], [5]), a suitable architecture is proposed to accommodate the faults occurred on p_{man}, *rpm* and *thr* sensors (hereinafter called respectively *x*, *y* and *z*) one at a time. For both ANN and SVR, it is composed of three structure (Ax, Ay, Az) with A={ANN, SVR}, each one able to accommodate faults on a specific sensor (*x, y, z*). Of course, only the ANN or SVR which corresponds to the faulty sensor is started one at a time. Each ANN and SVR (see Fig. 1) gives the correct value of the corresponding sensor, starting from the measured values of the other two sensors. As an example, if a fault has occurred on *x*, Ax provides the accommodated value of the *x* sensor (*xa*) on the basis of the quantities measured from *y* and *z* (faulty-free sensors).

Since different temporal dynamics of the involved measured quantities suggest to takes into account their time evolutions, in order to verify if the performance of the whole system are related to the availability of past samples, we have submitted to the prediction system the samples measured not only at the same instant *k* in which we want to evaluate the expected value of the sensor output, but also at instant *k-1, k-2, ..., k-d*, (*d = delay)* with *d=0,1,…,5* (for the sake of readability in section 5 we are only reported results for *d=0,3,5*) . For example *xa(k)*, that is the accommodated value if a fault has occurred at *k* instant on *x*, is generally expressed as:

$$xa(k)= f(y(k-d), y(k-d+1),...,y(k), z(k-d), z(k-d+1),...,z(k)) \tag{7}$$

that becomes $xa(k)= f(y(k), z(k))$ for *d = 0.*

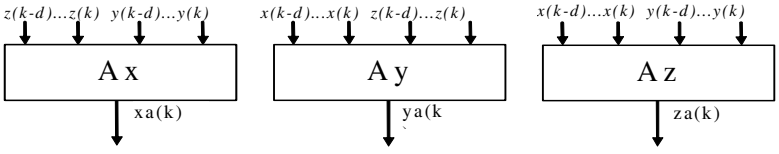


Fig. 1. The accommodation structure with A={ANN, SVR}

As a consequence, the dimension of each input vector (obtained in (7) with both *y* and *z*) of function *f* is *2(d+1)*. With similar architectures, the accommodated values for *y* and *z* (respectively *ya* and *za*) were obtained.

5 Experimental Results

As said in the previous section, the input data were organized in eight different sets of about 1100 samples. A preliminary pre-processing was carried out to constraint the input data in the range [0, 1]. Therefore, the samples were normalized by using:

$$inorm(k) = \frac{i(k) - \min(i)}{\max(i) - \min(i)} \quad \begin{cases} [\min(x), \max(x)] = [180,1000] \\ [\min(y), \max(y)] = [600,3600] \\ [\min(z), \max(z)] = [0,50] \end{cases} \tag{8}$$

where $i = x$ or y or z (the uppercase versions represents the normalized value), and *min(i)* and *max(i)* are the lower and upper limit, of the measured signals during typical engine work cycle. The *k-fold* cross validation technique (with $k = 8$) has been adopted in order to reduce the dependency from a particular data set (data set is *training*, *validation*, *test* set triple).

The prediction performance has been evaluated by using the mean absolute error (MAE) of the normalized data in the test set. Let MAE_j is the mean error evaluated on the *j-th* fold (containing $N_{k\text{-}fold}$ sample), with $j=1,\ldots,k$; as a global performance index we have adopted MAE_{mean}:

$$MAE_{mean} = \frac{1}{k}\sum_{j=1}^{k} MAE_j \text{ with } MAE_j = \frac{1}{N_{k-fold}} \sum_{m=1}^{N_{k-fold}} \left| Ia_m - I_m \right| \tag{9}$$

characterized by its standard deviation σ_{MAE} and by $\sigma_{MAE}\%$ defined as:

$$\sigma_{MAE}\% = 100\frac{\sigma_{MAE}}{MAE_{mean}} \tag{10}$$

The smaller the values of *MAE*, the closer are the predicted output to the actual values.

As for the ANNs, Multi Layer Perceptron schemes were adopted. All of them constituted of one input layer, one hidden layer, and one output layer. The activation function implemented in the hidden and output layers are, respectively, hyperbolic

tangent sigmoid transfer function and natural logarithm sigmoid transfer function. Several architectures were developed in terms of number of inputs, (spacing from 2 up to 12, depending on the delay chosen for the two input quantities), and number of hidden nodes (5, 10, 15 and 20). Matlab environment was adopted during the training and test phases. The learning sets are made of about 8800 samples acquired in different fault-free engine working conditions and the eight sets were so split: 6 as training set, 1 as validation set, 1 as test set. A Levenberg Marquardt back-propagation algorithm running on training set was used and validation set based early stopping was adopted.

The SVM module have been implemented by means of LIBSVM tool [11] (ver. 2.7) in Matlab environment. In the experimental phase [12], cross validation technique has been adopted to select the best kernel among *Polynomial, RBF,* and *sigmoid*. In this way, the *RBF* function has been chosen as the kernel function of SVR with parameters C (see (5)) and δ (see the end of section 3). The optimal association (δ, C) for each couple (d, I) (with $d=[0,3,5]$ and $I=\{X,Y,Z\}$) has been identified. After the training phase the *RBF* kernel obtained in the previous step has been tested with *k-fold* cross validation technique.

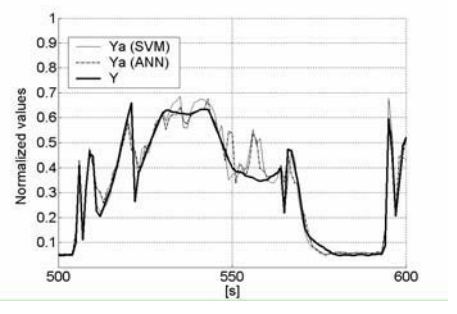


Fig. 2. An example of reconstruction: Ya is *rpm* accommodated (dashed and solid line are used for SVR and ANN respectively)

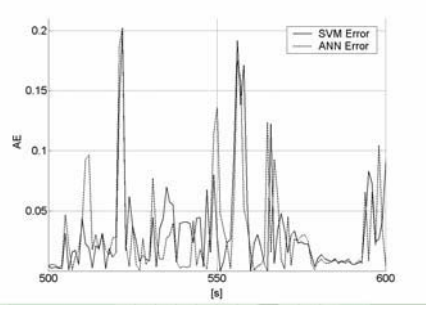


Fig. 3. Residuals (Absolute Error) relative to Ya reported in Fig. 2 (in the same time window)

As an example, in Fig. 2 and 3, the time evolutions of the actual value Y and of the Ya predicted by ANN and SVR, together with the corresponding absolute errors for a time window of the first test fold are reported. We can observe a similar behavior between the actual values and the predicted ones. Similar results were also achieved for the other test folds and in the prediction of the other quantities x and z. Later on, some synthetic performance indices, necessary to characterize quantitatively the systems, are reported.

For the sake of a global performance rating, the number of products needed by ANN or SVR during the prediction phase, have been evaluated. Let $N = 2(d+1)$ the dimension of inputs for both ANN and SVR, nSV the number of support vectors (for SVR), n_{hn} the number of neurons in the hidden layer (for ANN) we have about $P = (N+2)nSV$ products for SVR and $P=N(N_{hl}+2)+1$ products for ANN.

Table 1. Optimal SVR (δ, C) parameters for each d,I selected

		X	Y	Z
d	0	(0 .125, 2048)	(8.0, 2048)	(0.125, 2.0)
	3	(2, 2048)	(8, 128)	(0.125, 0.031)
	5	(0.125, 32)	(8, 8)	(0.125, 0.125)

Table 2. Results on test set for X

		ANN				SVM				
d	N	*MAE*	σ_{MAE}	σ_{MAE} %	*P*	ε	MAE	σ_{MAE}	σ_{MAE} %	L
0	5	2.397	0.1	4.1	15	0.200	8.243	0.3	3.7	98
	10	2.215	0.1	4.0	25	0.100	5.267	0.1	2.5	1207
	15	2.296	0.1	4.4	35	0.050	3.429	0.1	4.1	4350
	20	2.155	0.1	4.6	45	0.005	3.225	0.3	8.3	20888
3	5	2.451	0.2	6.1	57	0.200	8.583	0.3	3.7	149
	10	2.173	0.1	5.7	97	0.100	3.928	0.1	2.5	818
	15	2.072	0.1	4.6	137	0.050	2.765	0.1	4.1	4757
	20	5.857	3.9	***66.2***	177	0.005	2.246	0.3	***8.3***	***41110***
5	5	2.372	0.1	3.9	85	0.200	8.117	0.2	1.9	235
	10	5.553	3.0	54.3	145	0.100	3.983	0.1	3.4	1062
	15	1.914	0.1	3.3	205	0.050	***2.553***	0.1	2.5	5080
	20	***1.907***	0.1	6.0	***265***	0.005	3.709	0.1	4.7	28501

Table 3. Results on test set for Y

		ANN				SVM				
δ	N	*MAE*	σ_{MAE}	σ_{MAE} %	*P*	ε	MAE	σ_{MAE}	σ_{MAE} %	L
0	5	7.229	0.4	5.9	15	0.200	9.100	0.188	2.071	1417
	10	6.734	0.3	4.6	25	0.100	7.008	0.156	2.231	5016
	15	6.366	0.3	4.9	35	0.050	6.432	0.194	3.017	9693
	20	6.208	0.3	5.6	45	0.005	6.398	0.204	3.185	22563
3	5	5.681	0.2	3.1	57	0.200	8.351	0.799	9.566	1503
	10	5.486	0.5	8.3	97	0.100	5.606	0.634	11.304	5495
	15	4.994	0.3	6.2	137	0.050	7.050	0.259	3.667	12736
	20	4.798	0.4	7.8	177	0.005	4.278	0.558	***13.042***	41876
5	5	5.348	0.3	5.8	85	0.200	10.163	0.539	5.300	1843
	10	4.956	0.4	7.3	145	0.100	5.350	0.234	4.367	6773
	15	***4.465***	0.3	6.9	205	0.050	4.175	0.188	4.514	15143
	20	9.012	4.1	***45.2***	***265***	0.005	***3.746***	0.188	5.021	***53250***

Table 4. Results on test set for Z

		ANN				SVM				
d	N	*MAE*	σ_{MAE}	σ_{MAE} %	*P*	ε	MAE	σ_{MAE}	σ_{MAE} %	L
0	5	0.962	0.0	4.2	15	0.200	5.174	0.152	2.934	12
	10	0.962	0.0	4.2	25	0.100	4.862	0.136	2.805	44
	15	1.064	0.1	11.1	35	0.050	2.457	0.106	***4.331***	662
	20	0.910	0.0	5.3	45	0.005	1.771	0.056	3.173	***17459***
3	5	0.946	0.1	5.6	57	0.200	4.487	0.193	4.297	19
	10	1.994	1.2	***60.194***	97	0.100	3.187	0.070	2.182	65
	15	0.817	0.0	4.8	137	0.050	2.142	0.064	3.009	560
	20	3.327	1.7	49.8	177	0.005	1.517	0.027	1.799	16093
5	5	3.430	1.7	48.1	85	0.200	5.172	0.074	1.998	27
	10	0.868	0.1	9.6	145	0.100	3.247	0.073	2.305	84
	15	4.602	1.8	39.5	205	0.050	2.205	0.057	3.052	528
	20	***0.794***	0.1	7.2	***265***	0.005	***0.768***	0.028	2.174	15774

The obtained performance are showed in Tab. 2-4, where: δ is the RBF kernel parameter, C is an SVR training parameter, $d=delay$, $N=N_{hl}$ for ANN, $MAE= MAE_{mean}$ defined in (9), ε is the maximum absolute error accepted in training phase for SVR and finally P is the number of products for the prediction phase above defined.

The experimental results show that in terms of MAE_{mean} the SVR performance overcome the ANN for Y and Z sensors, whereas on X ANN furnishes better performance than SVR. Moreover, if we observe the σ_{MAE} and σ_{MAE}% trend we can see that the SVR behaviour is characterized by a smoothness that we can not observe for the ANN. The worst case for ANN takes place on X sensor: σ_{MAE}%$=66.2\%$ when $d=3$ and $N_{hl}=20$ while for SVR takes place for Y sensor: σ_{MAE}%$=13.0\%$ when $\varepsilon=0.005$ and $d=3$ (on the same sensor for ANN σ_{MAE}% $= 45.2\%$).

On the other hand, in order to obtain comparable performance with ANN the number of Support Vector (*nSV*) utilized by SVR in prediction phase is very large. For example to accommodate the Y sensor values with a *MAE* equal to *3.746,* with SVR must be evaluated *53250* products. A smaller amount of products (205) are necessary with ANN to obtain a *MAE* equal to *4.465*.

6 Conclusion

In this paper, the use of SVR compared with ANN in instrument fault accommodation scheme is verified. As a final results we have obtained that SVR has a comparable or better behavior with respect to ANN in terms of accuracy but not in terms of computational load. Used data set come from acquisition carried out on an engine test bench. Better performance for SVR are expected when greater generalization

capabilities are required such as in real vehicle runs where the regression task becomes more challenging since measured quantity dynamics depend on both vehicle inertia and very different drive conditions.

References

1. Dorr R., Kratz F., Ragot J., Loisy F., Germain J. L.:Detection isolation and identification of sensor faults in nuclear power plants. IEEE Trans. Contr. Syst. Technol, vol. 5, no. 1. (1997). 42-60
2. Chen J., Patton R. J., Liu J. P.: Detecting incipient sensor faults in flight control systems. Proc. Third IEEE Conf. Control Applications. Glasgow, U.K. (Aug. 1994). 871-876.
3. Betta G., Pietrosanto A.: Instrument fault detection and isolation: state of the art and new research trends. IEEE Transaction on Instrument and Measurement, vol. 49. (Feb. 2000) 100–107
4. Capriglione D., Liguori C., Pietrosanto A., Pianese C.: On-line sensor fault detection, isolation and accommodation in automotive engines. Instrumentation and Measurement, IEEE Transaction on, Vol. 52, Issue 4.(August 2003) 1182-1189
5. Capriglione D., Liguori C., A. Pietrosanto.: Real-time implementation of IFDIA scheme in automotive system. Proceedings of the 21-th IEEE Instrumentation and Measurement Technology Conference, Vol. 3. (May 2004) 1667-1672
6. Vapnik V.N., Lerner A.: Estimation of Dependences Based on Empirical Data. Springer-Verlag. Berlin. (1982) & Data Mining. Menlo Park, CA. (1995)
7. Scholkopf B., Burges C., Vapnik V.: Extracting support data for a given task. In Proc. First Conf. Knowledge Discover.
8. Muller R., Smola J.A., Scholkopf B.: Prediction time series with support vector machine in Proc Int. Conf Artificial Neural Networks. (1997) 999
9. Heywood J.B.: Internal Combustion Engine Fundamental. MC Graw Hill. (1988)
10. Mercer J. Functions of positive and negative type and their connection with the theory of integral equations. Philos. Trans. Roy. Soc. London, A 209: London. (1909).
11. Chang C.C., Lin C.J.: LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001)
12. Smola J.A., Scholkopf B.: A tutorial on support vector regression. Royal Holloway College. London U.K. NueroCOLT Tech. Rep. London. (1998)

Using Strings for On-line Handwriting Shape Matching: A New Weighted Edit Distance

Claudio De Stefano[1], Marco Garruto[2], Luis Lapresa[3], and Angelo Marcelli[2]

[1] Dipartimento di Automazione,
Elettromagnetismo Ingegneria dell'Informazione e
Matematica Industriale, Università di Cassino,
Via Di Biasio, 43 03043 Cassino (FR), Italy
destefano@unicas.it
[2] Dipartimento di Ingegneria dell'Informazione ed Ingegneria Elettrica,
Università di Salerno, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy
marcounisa@libero.it, amarcelli@unisa.it
[3] Departament de Ciències Matemàtiques i Informàtica,
Universitat de les Illes Balears, Carretera de Valldemossa,
Km 7,5 07122 Palma (Illes Balears), Spain
lapresa@gmail.com

Abstract. Edit Distance has been widely studied and successfully applied in a large variety of application domains and many techniques based on this concept have been proposed in the literature. These techniques share the property that, in case of patterns having different lengths, a number of symbols are introduced in the shortest one, or deleted from the longest one, until both patterns have the same length. In case of applications in which strings are used for shape description, however, this property may introduce distortions in the shape, resulting in a distance measure not reflecting the perceived similarity between the shapes to compare. Moving from this consideration, we propose a new edit distance, called Weighted Edit Distance that does not require the introduction or the deletion of any symbol. Preliminary experiments performed by comparing our technique with the Normalized Edit Distance and the Markov Edit Distance have shown very encouraging results.

1 Introduction

Edit Distance has been widely studied and successfully applied in a large variety of application domains. In fact, in the applications in which matching, detection or recognition of patterns are of primary interest, a key role is played by the way in which the similarity or the dissimilarity between patterns is measured: in this context, edit distance techniques offer an effective and computationally efficient way of performing such a measure, and it has been demonstrated that their applicability is not limited to alphabet-based strings in text processing, but they can be profitably used in a multitude of different applications. Examples include genome representation in bioinformatics [1], message codes in information theory [2] and sound information in speech processing [3]. Moreover, the concept of edit distance has been widely used in many research disciplines of pattern recognition, image processing and computer vision [4, 5].

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1125 – 1132, 2005.

The Edit distance between two strings was originally introduced in 1966 by Levenstein [6] and was defined as the minimum number of changes required to transform one string into another, where each change may be the insertion of a symbol, the deletion of a symbol or the substitution of a symbol with another. The greater the number of changes, the more different the strings are. In its original definition, called *Levenstein Distance* (LD), the changes are assumed to have the same (unitary) cost.

An obvious drawback of this definition is related to the fact that in real application the cost associated to a change may be different depending on the symbol inserted, deleted or substituted. To overcome this problem, a more general cost function has been introduced that allow to assign a different cost to each substitution of a symbol into another [7, 8]. The edit distance obtained in this way is called *Generalized Levenstein Distance* (GLD).

Another relevant problem encountered while using both LD and GLD is that such measures do not take into account the length of the patterns. This aspect may leads to meaningless measures in case of patterns having significantly different lengths: it is generally agreed that, for instance, two strings of length 4 having an edit distance equal to 2 should be considered much more dissimilar than two strings of length 100 having the same edit distance equal to 2. To solve this problem, Marzal and Vidal developed a *Normalized Edit Distance* (NED) in which the distance between two pattern is normalized by the length of the *edit path* [9]. In their work, they have also shown that NED outperforms any other normalization technique obtained by first computing the non normalized edit distance and then dividing by the length of the path.

Finally, to exploit the statistical dependencies among the values assumed by adjacent positions in the strings, an edit distance based on Markov Random Field, and therefore called *Markov Edit Distance* (MED), has been recently proposed [10].

Summarizing, the various edit distance techniques proposed in the literature differ for the way in which symbols to delete are selected, for the way in which both symbols to introduce and their position are chosen, and for the costs associated to each elementary operation (insertion, deletion and substitution).

Such techniques, however, share the property that, in case of patterns having different lengths, a number of symbols are introduced in the shortest one, or deleted from the longest one, until both patterns have the same length. This property, may have undesirable effects in case of application in which strings are used for shape description: in such cases, in fact, introducing or deleting a symbol may distort the corresponding shape resulting in a distance measure not reflecting the perceived distance between the shapes to compare.

Moving from these considerations, we propose a new edit distance, called Weighted Edit Distance (WED), based on the concept of *string stretching*: we do not introduce or delete any symbol in the strings to compare but simply extend or *stretch* the shortest string in such a way that each symbol of this string is compared with one or more symbols of the other, depending on the ratio r between the lengths of the two strings. The edit distance is then computed by summing the cost of substitution of each compared pair of symbols, weighted by a coefficient whose value depends on both the position of the symbols in the two strings, and the value of r. A normalization is also applied to overcome the problems previously discussed.

We have used WED in the framework of an on-line handwriting recognition system. In particular, we have developed an on-line handwriting segmentation method [11, 12] which allow to extract the elementary strokes a word is composed of, and provides segmentation points that are very accurate and stable, i.e. very invariant with respect to non significant variations in the shape of the ink. Each elementary stroke has been described by computing the arclength [13] representation and quantizing the values of the angles into 16 intervals encoded by one of the letter in the subset [A-P]. By this encoding, the shape of each elementary stroke is described by a string of characters.

Preliminary experiments performed by comparing our technique with the Normalized Edit Distance and the Markov Edit Distance have shown very encouraging results.

The remaining of the paper is organized as follows: Section 2 illustrates our method for on-line handwriting shape description. Section 3 describes the proposed Weighted Edit Distance technique. Preliminary experimental results and some concluding remarks are eventually left to Section 4.

2 The Shape Description Method

Studies on visual perception have shown that curvature plays a key role in our perception of shape and its organization into parts [14]. Therefore, since the sixties many efforts have been made to develop algorithms for computing the curvature along a line and then use this information for both locating curvature maxima, in order to extract the elementary parts forming the shape, and describing the shape of each part by some encoding of its curvature [15].

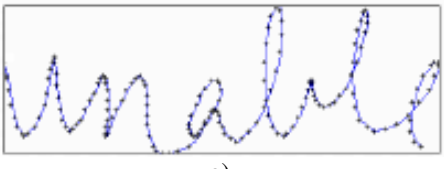
Unfortunately, applying the mathematical definition of curvature, may result in detecting many curvature maxima not corresponding to perceptually relevant points, eventually providing different descriptions for similar shapes.

To solve this problem, we developed a method derived from an analogy with the primate visual system for selecting the best scale at which the electronic ink of the handwriting should be described [12]. According to this analogy, the method computes a multiscale features maps by evaluating the curvature along the ink at different levels of resolution and arranges them into a pyramidal structure. Then, feature values extracted at different scales are combined in such a way that values that locally stand out from their surrounds are enhanced, while those comparable with their neighbours are suppressed. A saliency map is eventually obtained by combining those features value across all possible scales. Such a map is then used to select a representation that is largely invariant with respect to non significant shape variations encountered in handwriting.

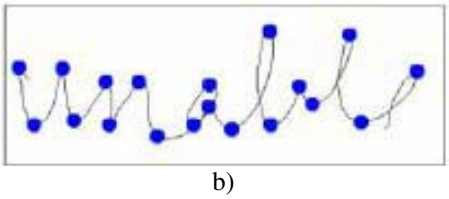
Once the appropriate scale has been selected, the arclength representation [13] of the electronic ink at that scale is considered. This representation is a function $\alpha(\lambda)$ where λ is the curvilinear abscissa of a point, and α is the angle of the tangent to the curve at that point with respect to the horizontal axis. The segmentation of the word into elementary strokes is performed by locating the curvature maxima on the arclength representation at the selected scale. The final description of the handwriting shape is given in terms of a set of strings each encoding the curvature changes relative

to each elementary stroke. To this aim, the actual values of $\alpha(\lambda)$ are quantized into 16 intervals and each interval encoded by one of the letter of the subset [A-P] in such a way that the letter A corresponds to the first interval (from 0 to $2\pi/16$), the letter B to the second one (from $2\pi/16$ to $2*2\pi/16$) and so on, counter clockwise. By this encoding, the shape of the word is described by a string of characters that represents the desired set of features.

Figure 1 reports the electronic ink of a word, the segmentation points that allow to split the original word into elementary strokes and the strings encoding each elementary strokes.



a)



b)

J-MMNO-CDDEC-MMNO-CDD-NMM-ADDC-NMMNP-BBCDE-JKLN-CA-NP-
CCDDDEE-HLMMMMNO-CDD-OO-BCDDDE-LLMMMNN- ABCCDF-KLLL

c)

Fig. 1. a) The original input. b) The segmentation of the ink in elementary strokes. c) The sequence of strings encoding the change of curvature along the curve at the selected resolution: hyphens correspond to segmentation points.

3 Weighted Edit Distance (WED)

The basic idea the Weighted Edit Distance is based upon is that of avoiding the introduction or the deletion of symbols in the strings to compare by *stretching* the shortest string in such a way that each of its symbol is compared with one or more symbols of the other, depending on the ratio *r* between the lengths of the two strings.

To formally describe the algorithm for computing WED, let us denote with S1 and S2 the two strings to compare, with S1[i] (S2[j]) the i-th (j-th) symbol in the string S1(S2), and with Ls1 and Ls2 the lengths of S1 and S2, respectively. Let us also assume that $Ls1 \geq Ls2$. Finally, let us denote with D(x,y) the function that assigns a cost to the substitution of symbol x with symbol y. In this study D(x,y) is equal to the alphabetical distance between x and y, i.e. D(A,B)=1, D(A,C)=2, and so on.

The algorithm for computing WED is the following:

```
Program WED
var     Ls1, Ls2    : integer;
        S1,S2       : string;
        r           : real;
        overlap_S1  : real;
        overlap_S2  : real;
        weight      : real;
        dist        : real;
        i,j         : integer;
begin
    get strings S1 and S2 and their lengths
    dist:=0;
    if (Ls1 = Ls2) then
        for i:=1 to Ls1 do
            dist := dist + D(S1[i],S2[i]);
    else begin
        r:=Ls1/Ls2;  overlap_S1:=1;  i:=0;  j:=1;
        while (i<Ls2) AND (j≤Ls1) do begin
            i:=i+1;  overlap_S2:=r;
            while (overlap_S2>0) do begin
                weight := min(overlap_S1,overlap_S2);
                dist := dist + D(S1[i],S2[j]) * weight;
                overlap_S1 := overlap_S1 - weight;
                overlap_S2 := overlap_S2 - weight;
                if (overlap_S1 ≤ 0)then begin
                    j := j + 1;
                    overlap_S1 := 1;
                end
            end
        end
    end
    dist := dist / Ls1;
end.
```

The variable overlap_S1 dynamically indicates the percentage of the current symbol in S1 which has not been exploited so far in the comparison with the symbols in S2. The values assumed by this variable, obviously range from 1 to zero.

The variable overlap_S2 dynamically indicates if the current symbol in S2 has been completely exploited in the comparison with the symbols in S1. The values assumed by this variable, range from *r* to zero. When the variable overlap_S2 assumes a value greater than or equal to one, it indicates that there are more symbols in

S1 that must be compared with the current symbol in S2: as a consequence, the value of overlap_S1 is assigned to the variable weight, the current symbol in S1 is compared with the current symbol in S2 and then the next symbol in S1 is selected. When the variable overlap_S2 assumes a value less than one, such a value represents the residual percentage of the current symbol in S2 that has not been exploited. It also indicates that the current symbol in S1 is the last one that must be compared with the current symbol in S2. In this situation the value of overlap_S2 is assigned to the variable weight, the current symbol in S1 is compared with the current symbol in S2 and then the next symbol in S2 is selected.

Summarizing, the edit distance is computed by adding at each step the cost of substitution of the current pair of symbols multiplied by the current value of the weight: this process is iterated until both symbols in S1, and symbols in S2 have been completely exploited. Finally the value of the distance is normalized by Ls1.

4 Experiments and Discussion

In order to validate the proposed edit distance technique, a set of experiments have been performed for comparing our results with those obtained by using the Normalized Edit Distance and the Markov Edit Distance. A further experiment has been eventually performed to show some preliminary results obtained in the framework of our on-line handwriting recognition system.

The aim of the first set of experiments is that of evaluating how WED *is continuous with respect to the perception*, i.e. how the distance variations measured by WED are consistent with the perceived entity of the modifications as the shape of an object is changed smoothly. Figure 2 illustrates the shape we have used for this experiments, while figure 3a)-c) plots the distance between one of the shape shown in figure 2 (shape 1, 4 and 7, respectively) and all the other ones by using WED, NED and MED. For the sake of comparison, all the shapes are composed by the same number of points and consequently are described by strings having the same length. The plots show that WED always exhibits a very consistent behavior with the variation of shape. In particular, its linear trend reflects the perceived continuous variations among the considered shapes. In contrast, NED is linear in the plot of figure 3a) , but not in that of figure 3c), that has been obtained by considering the distances among the shape in the reverse order with respect to that of figure 3a). MED exhibits only a

Fig. 2. The “continuum” of shapes used in the first experiment

partial linearity, in that for some shapes there are no difference in the measured distance. Eventually, NED is non symmetric when the same amount of changes occurs on both side of the reference shape.

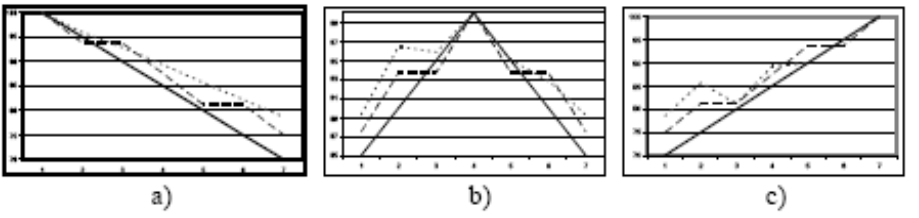


Fig. 3. The distances among the shapes of figure 2 with repect to: a) shape 1; b) shapes 4; c) shape 7. In the plots, continuous lines represent WED, dashed lines MED, dotted lines NED. a)

In the second experiment, we have considered 100 words from a lexicon containing pairs of words sharing at least one bigram or trigram. The words, extracted from a database of cursive words produced in our lab, were processed and the distances among the obtained strings measured according to WED. The strings belonging to different words and corresponding to the minimum distances were eventually concatenated to obtain the best common sequence of strings. Figure 4a) shows the ink corresponding to such a sequence for two fragments of the words "ceramica" (*ceramic*) and "cerato" (*waxed*), while figure 4b) plots the distances among the corresponding strokes of the two words. As the figure illustrates, the best common sequence of strings corresponds to the most similar fragments of ink.

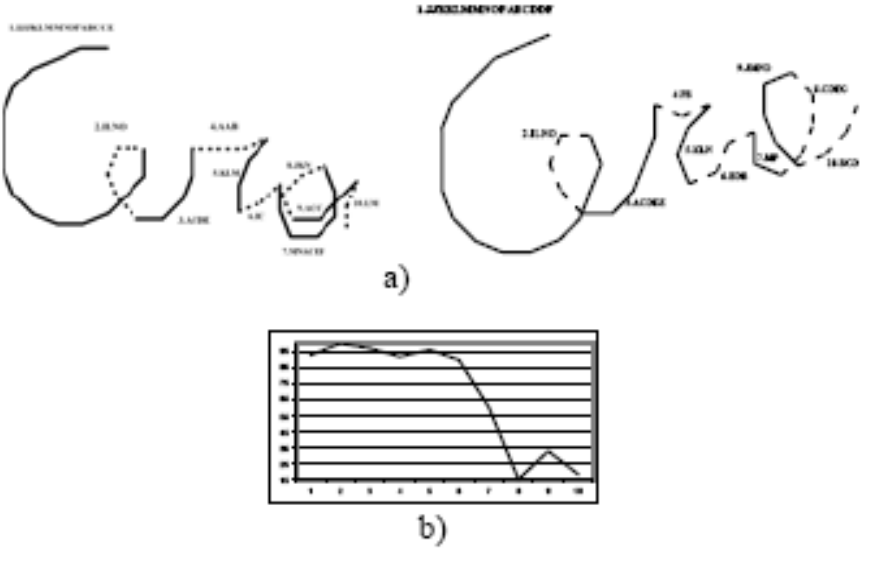


Fig. 4. The application of WED to handwriting shape. a) Ink strokes and the corresponding strings; b) high values correspond to similar shapes (strokes 1 to 6)

The experiments reported here suggest that the proposed distance is a promising tool toward a quantitative measure of the elusive concept of shape similarity. Further experiments on a larger database, together with a formal assessment of the metric properties of WED, are however needed in order to confirm the behavior observed during the experiments performed in this study.

References

1. Durbin, K., Eddy, S., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge Univ. Press (1997).
2. Cover, T.M., Thomas, J.A.: Elements of Information Theory, second ed. Wiley (1991).
3. Huang, X., Acero, A., Hon, H.: Spoken Language Processing. Prentice Hall (2001).
4. Tsay, Y.T., Tsai, W.H.: Attributed String Matching Split-and- Merge for On-Line Chinese Character Recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 15, no. 2 (1993), 180-185.
5. Ratha, N., Chen, S., Karu, K., Jain, A.K.: A Real-Time Matching System for Large Fingerprint Databases, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18, no. 8 (1996) 799-813.
6. Levenstein, A.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals, Soviet Physics-Doklandy, vol. 10 (1966).
7. Okuda, T., Tanaka, E., Kasai, T.: A Method of Correction of Garbled Words Based on the Levenstein Metric, IEEE Trans. on Computers, (1976), 172-177.
8. Wagner, R.A., Fisher, M.J.: The String to String Correction Problem, Journal of ACM, vol. 21, no. 1 (1974), p. 168-173.
9. Marzal, A., Vidal, E.: Computation of Normalized Edit Distance and Applications, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 15, no. 9 (1993), 926-932.
10. 10. Wei, Jie: Markov Edit Distance, IEEE Trans. on Pattern Analysis And Machine Intelligence, vol. 26, no. 3 (2004), 311-321.
11. De Stefano, C., Guadagno, G., Marcelli A.: A Saliency-Based Segmentation Method for On-Line Cursive Handwriting, International Journal of Pattern Recognition and Artificial Intelligence, vol.18, no. 6 (2004), 1139-1156.
12. De Stefano, C., Garruto, M., Marcelli, A.: A saliency-based multiscale method for on-line cursive handwriting shape description", Proc. of the 9th International Workshop on Frontiers in Handwriting Recognition 2004 (IWFHR-9), Tokyo, Japan, October, 26-29, 2004, 124-129.
13. Pavlidis, T.: Structural Pattern Recognition, Springer-Verlag, 1977.
14. Fischler, M.A., Bolles, R.C.: Perceptual organization and curve partitioning, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 8, no. 1 (1986), 100-105.
15. Marshall, S.: Review of shape coding techniques, Image and Vision Computing, vol. 7, no. 4 (1989), 281-294.

Automatic Updating of Urban Vector Maps

S. Ceresola, A. Fusiello, M. Bicego*, A. Belussi, and V. Murino**

Dipartimento di Informatica, Università di Verona
Strada Le Grazie 15, 37134 Verona, Italy

Abstract. In this paper we propose an automatic updating system for urban vector maps that is able to detect changes between the old dataset (consisting of both vector and raster maps) and the present time situation represented in a raster map. In order to automatically detect as much changes as possible and to extract vector data for new buildings we present a system composed of three main parts: the first part detects changes between the input vector map and the new raster map (based on edge matching), the second part locates new objects (based on color segmentation), and the third part extracts new objects boundaries to be used for updating the vector map (based on edge detection, color segmentation and adaptive edge linking). Experiments on real datasets illustrate the approach.

1 Introduction

A significant amount of work has been done in the field of aerial image processing in particular regarding the detection of buildings or other man-made structures such as roads, railways, and others [1,2,3,4,5]. These results could be of great help, if they were applied to update data stored in a Geographical Information System (GIS).

Geographical data in a GIS are usually structured in different logical layers, each one containing a specific type of geographical features [6] (for example, there could be a layer for roads, one for buildings and one for rivers). We call this set of layers a *GIS database*. Data in layers are obtained from survey processes that require: a) the acquisition of aerial images of the territory of interest and b) the processing of these images by human experts using specific optical tools in order to extract the vector representation of the features of interest. GIS applications require up to date information in order to be effective, and this cannot be achieved without time consuming and expensive operations, if the above approach is applied; indeed, it requires to repeat the process from the beginning creating a completely new layer of vector data.

As suggested by some previous papers in the areas of GIS and remote sensing [7,8], a layer of a GIS database, containing the vector representation of a set of

* Current Address: DEIR, Università di Sassari - via Torre Tonda 34, 07100 Sassari, Italy.

** Corresponding author: Tel +39 045 802 7996, Fax +39 045 802 7068, E-mail vittorio.murino@univr.it

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1133–1139, 2005.

features, can be updated by comparing the original aerial image (from which the vector representation was derived) against some new images of the same area. Now, the problem is: how can we reduce the cost of this operation? Obviously the cost for the realization of the new ortophotos cannot be avoided, however we can reduce the cost of the comparison operation between old and new data by adopting automatic tools that are able to automatically detect changes and update datasets avoiding human work. In this paper an automatic approach to GIS updating is proposed, based on image processing techniques, namely edge matching, color segmentation and edge detection and linking.

1.1 Overview of the Proposed Approach

Our approach is divided in three phases. In the following, we call $V_{old}(L)$ the old vector representation of the layer L, R_{old} the old ortophoto, or old raster map, and R_{new} the new ortophoto, or new raster map. Moreover, we focus on features with a polygonal representation in vector format.

1. *Detection of feature boundary changes*: for each new raster map R_{new} and for each layer L, we identify the known features of L on R_{new}, by superimposing V_{old} onto R_{new}. By comparing the edges of known features, and the ones obtained by a gradient analysis on R_{new}, we detect changes of old features. These changes are further validated by comparing R_{new} against R_{old} with a robust change detection technique. As a byproduct, we obtain a color range for each feature type by considering all the areas of R_{new} where changes have not been detected.
2. *Finding location of new features*: for each new raster map R_{new} and for each layer L we identify new features of the type represented in layer L, by classifying R_{new} using the color ranges stored for layer L.
3. *Recovery of vector data for new features and update of vector map*: for each new raster map R_{new} and for each layer L the edges of the new features identified by the previous phase are computed and stored in $V_{new}(L)$.

In this paper a direct application of this general approach to the buildings layer updating is proposed: preliminary experimental evaluation on real datasets shows promising results.

In the paper we analyze in detail each part of the system. In particular, Section 2 presents the proposed approach for automatic detection of changes and contains a subsection for each phase of the approach as above described. Section 3 illustrates the results obtained by apply this approach to real aerial images and GIS databases. Finally, Section 4 outlines conclusions and future perspectives.

2 Method

2.1 Detection of Buildings Boundary Changes

The method we used is inspired by [9]. The input data of the system are: the vector map at time t (V_{old}), the raster image at time t (R_{old}) and the raster

image at time $t + \Delta t$ (R_{new}), where Δt is the time lag between the old and the new image.

We need to compare old information with the newer one and to identify changes that occurred in the meantime. In our approach, this is equivalent to comparing an object as it is represented in an image captured at instance $t + \Delta t$ to the same object represented in a vector format at time t. In the case of buildings, the vector data represents the building outline.

The old vector information is retrieved from a GIS database. Using positional information from this database and geo-referencing information from the new image, we transfer the old object information onto the new image. V_{old} produces edge templates that should be validated in R_{new}.

To make the comparison more precise, we partition our vector polygons in smaller segments, according to the MSE (Minimum Spatial Element) criterion defined in [10], which defines the spatial resolution of changes.

We locate the gradient maximum value in the direction perpendicular to the edge template: we compare the position of this maximum value with the edge template position. If the difference is less than few pixels, vector data is validated, otherwise a possible change is detected. Before a change flag is raised, the detection must be confirmed on the raster image with a robust method for change detection based on the following ratio:

$$\lambda = \frac{\left[\frac{\sigma_{old} + \sigma_{new}}{2} + \left(\frac{\mu_{old} - \mu_{new}}{2}\right)^2\right]^2}{\sigma_{old} * \sigma_{new}} \tag{1}$$

where μ and σ are the mean gray value and the variance for the region being compared. Only if λ is greater than a threshold, a change is detected.

This copes with the case when an edge of V_{old} does not match with R_{new}, but no changes happened. Consider for example the case when vegetation occludes edges both in R_{new} and R_{old}.

In short, we can resume step-by-step our method as follow:

1. Transferring the old vector information onto the new image.
2. Decomposing the outline (old vector data) according to the defined MSE.
3. Calculating sum of gradient in the direction perpendicular to the edge.
4. Calculating peaks of gradient and their position.
5. Selecting the greater peak and comparing its position with that of edge.
6. Applying the robust change detection method.

Please note that this part of our system is only able to detect changes in building already present in V_{old}. Namely, the following situations are catered for: buildings that have been enlarged/reduced, or building that have been pulled down.

The following section describe our strategy for locating *new* buildings, i.e., buildings constructed after time t.

2.2 Finding New Buildings' Location

Observing a lot of aerial images, we can notice that buildings' rooftop in the same area have a good probability to have a similar color. We can therefore define a color range for buildings by computing the average color value of areas where no changes have been detected. This is particularly important since it allows an automatic definition of the range based on the present time image R_{new}.

The method for detecting new buildings is based on simple color image segmentation. As customary, we work in the HSV (Hue Saturation and Value) color space and use the chrominance (hue and saturation) information only.

The image is segmented by thresholding the H and S image channels. The threshold values $\theta_{\min}$ and $\theta_{\max}$ (one for each channel) are computed as follows:

$$\theta_{\min} = \mu - c, \qquad \theta_{\max} = \mu + c \tag{2}$$

where μ and c represents the color range obtained from examples.

The result is then refined with morphological operations [11]. We can notice that buildings have generally a regular, and compact shape. By analyzing ratio between perimeter and area of blobs in the output binary image, we can select the areas that are reasonably roofs.

2.3 Recovery of Vector Data for New Buildings and Update of Vector Map

In this last part of the system, the vector map V_{old} is updated to obtain the new map V_{new}. To this end we take into account the parts of V_{old} that have not been validated (Sec 2.1), corresponding to changes to old buildings, and the new buildings detected in Sec. 2.2.

Let us start with the processing applied in the case of new buildings. The output of the previous stage is the geo-referenced location of areas containing new buildings. The goal is to obtain a polygonal representation for each building. The boundary of the building are located using both edge detection (Canny [12] operator) and the result of color segmentation from the previous phase.

The found edges and regions are then integrated using an adaptive method that fits a closed polygon. This approach has some advantages:

- it is a very natural way of combining edge detection and color segmentation results;
- it enforces the type of result that we are looking for, namely closed polygons;
- it does not impose unnecessary constraints on the shape a building could have.

Our fitting method starts with a bounding box, strictly containing the segmented region, decomposed according to the method used in Section 2.1. Then, the first segment is moved toward the center of the region until it reaches the boundary of the building. This condition is expressed by requiring that the segment should coincide with as much edge points as possible and should lie on a

color boundary. In formulae, the following two conditions must be satisfied. The first is

$$\frac{1}{n}\sum_{i=1}^{n} E(x_i, y_i) > \lambda_1 \tag{3}$$

where $i \in [0, n]$ with n number of points in the segment, x_i is the abscissa of i-*th* segment point and y_i its ordinate, E the edge map produced by edge detection process and λ_1 is a threshold. The measure above express the rate of segment points that coincide with edge points. The second writes:

$$s_{i+1} \cong 0 \quad \text{and} \quad s_{i-1} \geq \lambda_2 \tag{4}$$

where s_{i+1} is the rate of points just one pixel above the segment that belongs to the segmented region, and s_{i-1} is the rate of points just one pixel under the segment that belongs to the segmented region.

When such a position is reached, the next segment is placed with the additional constraint that one endpoint must coincide with the previous, so as to form a closed polyline.

In the case of updating the representation of an already exiting building, only the region where the changing took place is considered, and only segments that have not been validated are moved.

(a) (b)

Fig. 1. (a) Historical image; (b) present time image

3 Results

The proposed method has been tested using aerial images of the Davies city, California[1]. The images were digitized and orthorectified to remove distortion introduced from varying camera angles and distances. These images are taken as R_{old}, whereas the present images R_{new}, are obtained by manually editing the old ones.

Fig. 2. Result: yellows parts are confirmed vector data, the blue parts are new buildings not present in the vector data (i.e. not present in the historical images) and the red parts are the parts described in the vector data but that do not match with the present image.

An example of two images taken in input is shown in Figure 1(a) and (b), representing the old image R_{old} and the present image R_{new}, respectively.

Comparing the two images one can see that a house located in the bottom left of the image disappears in the present image, while another house appears in the bottom left of the central agglomerate of houses. A littler difference could also be noticed in the first and the fifth house of the second row from the top, which change their configuration. The vector data taken in input is relative to the historical image, and the goal is to update it basing on the new image, *i.e.* to find the disagreements described below. The result obtained with the application of our method is presented in Figure 2: yellows parts are confirmed vector data, the blue parts are new buildings not present in the vector data (i.e. not present in the historical images) and the red parts are the parts described in the vector data but that do not match with the present image.

[1] Available at `http://www2.dcn.org/orgs/orthophotos.`

One can note that obtained results are really satisfactory, all disagreements between older and new images have been found and coded in polygonal form, suitable for vector data updating.

4 Conclusions

In this paper we proposed a complete semi automatic method to update vector maps of a GIS database. Detection of changes regarding a given feature type (i.e., roads, buildings, etc.) is performed by integrating vector and raster data. Features are located based on color segmentation and shape; moreover, vector outline of new features are extracted through an new adaptive edge linking method. The proposed method has been applied to a layer containing buildings and it has been tested on real high resolution images, presenting encouraging results.

References

1. Meisels, A., Bergman, S.: Finding objects in aerial photographs: a rule-based low level system. In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition. (1988) 118–122
2. Mayer, H.: Automatic object extraction from aerial imagery: A survey focusing on buildings. Computer Vision and Image Understanding **74** (1999) 138–149
3. Auclair-Fortier, M.F., Ziou, D., Armenakis, C., Wang, S.: Survey of Work on Road Extraction in Aerial and Satellite Images. Technical Report 247, Département de mathématiques et d'informatique, Université de Sherbrooke (2000)
4. Bicego, M., Dalfini, S., Murino, V.: Extraction of geographical entities from aerial images. In: Proc. of IEEE Workshop on Remote Sensing and Data fusion over Urban Areas (URBAN03). (2003) 125–128
5. Bicego, M., Dalfini, S., Vernazza, G., Murino, V.: Automatic road extraction from aerial images by probabilistic contour tracking. In: Proc. of IEEE Int. Conf. on Image Processing. Volume 3. (2003) 585–588
6. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases with Application to GIS. Morgan Kaufmann Publishers (2002)
7. Walter, V., Fritsch, D.: Automated revision of gis databases. In: Proc. of the ACM GIS 2000. (2000) 129–134
8. Walter, V.: Automatic classification of remote sensing data for gis database revision. International Archives of Photogrammetry and Remote Sensing **XXXII** (1998) 641–648
9. Agouris, P., Monuntrakis, G., Stefanidis, A.: Automated spatiotemporal change detection in digital aerial imagery. In: Proc. of SPIE Aerosense2000, Orlando, FL. (2000)
10. Mountrakis, G., Agouris, P., Stefanidis, A.: Navigating through hierarchical change propagation in spatiotemporal queries. In: Proc. of the IEEE Time 2000 Workshop. (2000) 123–131
11. Castleman, K.: Digital Imgage Processing. Prentice Hall (1996)
12. Canny, J.: A computational approach to edge detection. IEEE Trans. on Pattern Analysis Machine Intelligence **8** (1986) 679–698

An Autonomous Surveillance Vehicle for People Tracking

C. Piciarelli, C. Micheloni, and G.L. Foresti

Department of Computer Science, University of Udine,
Via delle Scienze 206, 33100 Udine, Italy

Abstract. In this paper, the problem of the surveillance and the security of indoor environments is addressed through the development of an autonomous surveillance vehicle (ASV). The ASV has been designed to perform the object detection by adopting an image alignment method followed by a change detection operation. Hence, in addition to the classical robotic tasks (e.g., navigation and obstacle avoiding), the tracking of objects (e.g., persons) moving in an indoor environment is considered. The tracking procedure allows the ASV to maintain the interesting objects in the centre of the image, and in specific cases to focus the image acquisition on particular parts of the object (e.g., face of a person, etc.) for recognition purposes. Experimental results have been performed on different real scenarios where no objects moves inside the monitored scene and where at least one moving object is into the scene.

1 Introduction

The research work done in the last years on the field of mobile vision focused on many fields: aircrafts [2], autonomous underwater vehicles (AUV) [11] and autonomous guidance vehicles (AGV) moving on the ground [8, 13]. Certainly the work made in guidance of ground vehicle represents the great part of the research in context of mobile navigation where great interest has been pointed on indoor environments. From the first systems like those proposed by Giralt et al. in [6] and by Moravec in [10] many systems have been developed. For greater details on indoor navigation systems the reader is referred to [4]. Recently, some researchers have analyzed the problem of employing AGVs for surveillance purposes in both outdoor and indoor environments. In [3], Lipton et al. employed an airborne platform and ground vehicles to develop a multicamera system to monitor activities in cluttered outdoor environments using a distributed network of fixed and mobile sensors. In this context, the employment of mobile robots equipped with specific visual sensors for surveillance purposes can become an important tool.

In this paper, we have focused our attention to the problem of the surveillance and the security of indoor environments and, to address this objective, an autonomous surveillance vehicle (ASV) has been designed and developed. The ASV is able to perform, in addition to the classical robotic tasks (e.g., navigation in an indoor environment by avoiding obstacles), the tracking of mobile objects (e.g., persons). The selection of the target object to be tracked can be decided by a remote operator or autonomously by the system itself in the case of the presence of an alarm. The tracking procedure allows the system to maintain the objects of interest in the centre of the image, and in specific

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1140–1147, 2005.

cases to localise particular parts of the object (e.g., face of a person) for recognition purposes.

The proposed ASV is able to detect moving objects by means of a direct method [14] and by computing the affine transform for the alignment of the two consecutive frames. An iterative and a multiresolution alignment techniques have been interleaved to increase the reliability of this task. Moreover, to overcome the problem of parallax an active zoom based technique has been introduced to increase the alignment performance in hallways context. Once the objects have been detected, a tracking module [5], based on a Kalman Filter, is able to estimate the position of the objects at the next time instant. Such estimations are used by the control module to autonomously change the position of the ASV in order to maintain the objects of interest as close as possible to the centre of the image.

2 Motion Detection

The principal activity of this module is the detection of mobile objects inside the scene. The problem consists in the identification of the motion due to the camera actions and of the real motion of the objects. The proposed solution is given by applying a direct method [14] to compute the affine transform for the alignment of two consecutive frames.

From the equation of the optical flow [9] and by considering the equation of the affine displacements, the following equation of the affine flow holds:

$$I_x(a_{11}x + a_{12}y + a_{13}) + I_y(a_{21}x + a_{22}y + a_{23}) + I_t \tag{1}$$

where I_x,I_y and I_t are the spatial and temporal derivatives of the brightness intensity.

From (1), it is possible to define a linear system $\mathbf{Ax} = \mathbf{b}$ by considering each pixel of the image. Therefore, in order to reduce its computational complexity, we have adopted the form $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$ which yields to (2) where only the upper section of the symmetric matrix $\mathbf{A}^T\mathbf{A}$ has been shown.

$$\underbrace{\begin{bmatrix} \sum I_x^2x^2 & \sum I_x^2xy & \sum I_x^2x & \sum I_xI_yx^2 & \sum I_xI_yxy & \sum I_xI_yx \\ & \sum I_x^2y^2 & \sum I_x^2y & \sum I_xI_yxy & \sum I_xI_yy^2 & \sum I_xI_yy \\ & & \sum I_x^2 & \sum I_xI_yx & \sum I_xI_yy & \sum I_xI_y \\ & & & \sum I_y^2x^2 & \sum I_y^2xy & \sum I_y^2x \\ & & & & \sum I_y^2y^2 & \sum I_y^2y \\ & & & & & \sum I_y^2 \end{bmatrix}}_{\mathbf{A}^T\mathbf{A}} \underbrace{\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{bmatrix}}_{\mathbf{x}} = \underbrace{-\begin{bmatrix} \sum I_tI_xx \\ \sum I_tI_xy \\ \sum I_tI_x \\ \sum I_tI_yx \\ \sum I_tI_yy \\ \sum I_tI_y \end{bmatrix}}_{\mathbf{A}^T\mathbf{b}} \tag{2}$$

Adopting such equation for the affine transform computation, requires to solve the problems involved by the optical flow. First, when both spatial derivatives are equal to zero an adaptive thresholding technique [12] allows to discharge the 80% of the pixels by considering only those whose gradient is greater than a threshold T_{pg} ($\nabla I(x, y) \geq T_{pg}$). Instead, to sidestep the inability of the optical flow method to work on large displacements, we have adopted two techniques: a) Iterative alignment and b) Multiresolution alignment. Two heuristics have been studied to speed up the iterative alignment process compared to the Newton-Raphson scheme. The first consists in the factorization of the

equation (2) by adopting the method proposed by Cholesky. This resulted in a reduction of the complexity of a factor 2. The second is given by introducing a time correlation of the alignment between two consecutive frames. Precisely, the iterative process is initialised with the parameters of the affine transform computed for the previous frame. The proposed solution with the adopted heuristics is presented in Algorithm 1.

Algorithm 1 Iterative Alignment

$\mathbf{a}(0) = \mathbf{a}_{old}$
$\mathbf{L} = Cholesky(\mathbf{A}^T\mathbf{A})$
$t = 1$
repeat
  $I(t) = AffineTransform(I, \mathbf{a}(t-1))$
  $Calculate\mathbf{A}^T\mathbf{b}$
  $Solve(\mathbf{L}\mathbf{L}^T\Delta\mathbf{a} = \mathbf{A}^T\mathbf{b})$
  $\mathbf{a}(t) = \mathbf{a}(t-1) + \Delta\mathbf{a}$
  $t = t + 1$
until $\Delta\mathbf{a} < Thresh$

Since the iterative method, adopted in context of wide displacement between consecutive frames, requires an high number of iterations, we propose an alignment method based on a multiresolution approach. Each frame at different resolutions, determined by using a logarithmic scaling factor,is considered. The parameters of the affine transform are computed first at the highest level (i.e. the level corresponding to the image with lower resolution) then on the lowers. At each level a new iterative process for the alignment computation is started by using the partial solution reached at the previous level.

Besides this, optical flow implies to overcome a further problem given by the computation of the affine flow on pixel belonging to moving objects. Precisely, exploiting such pixels in the computation of the affine transform would result in a wrong alignment. Hence, in order to reduce their affect on the result or to give them a minor weight into the equation (2), we have adopted a robust estimator. The choice of the iteratively reweighed least square (IRLS) [1] as robust estimator, allows to maintain the structure of the described method. With this estimator the linear system in equation (2) becomes:

$$w\mathbf{A}^T\mathbf{A}\mathbf{x} = w\mathbf{A}^T\mathbf{b} \tag{3}$$

where the weights w_j are computed by the estimator as follows:

$$w_i = w(x_i) = \frac{\rho'(x)}{2x_i} \tag{4}$$

where ρ is the function of the estimator and x_i is the residual of the pixel i. The function selected as estimator is the

$$\rho(x, \sigma) = log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right) \tag{5}$$

with σ equal to the scaling factor.

Fig. 1. People detection inside a group. First column: original sequence, second column: the result of the region identification algorithm, third column: face and body detection, fourth column: bounding ellipses for each person composing the group.

Once the alignment phase has been successfully executed, a change detection operation is applied on the two considered frames in order to detect the moving pixels. The resulting binary image is processed to compute the bounding ellipses of all moving blobs.

If the detected blob is composed by a group of people, in order to segment different persons a separation algorithm was developed. The proposed method searches the faces inside the blobe using a face detector based on a cascade of boosted classifiers working with Haar-like features, as described in [16]. The moving blob is then segmented in different regions of similar colours, and finally the regions are associated to the nearest detected face in order to identify the body of each person. The whole process is shown in figure 1.

3 Active Zoom

In this section it will be discussed how the zoom control affects the proposed ASV system. Before studying how the ASV performances can be improved by the active control of camera zoom, it must be shown that the proposed system can control it. Actually many tracking systems cannot use the zoom because of the algorithms used for image alignment (e.g., it is the case of correlation-based systems) since correlation is not scale independent.

In the simple pinhole camera model, zoom is modelled as the change of the distance of the image plane from the pinhole. The zoom thus moves each pixel at coordinates (x, y) in the new position (x', y'), where

$$(x, y) = \left(f\frac{X}{Z}, f\frac{Y}{Z} \right) \qquad (x', y') = \left(f'\frac{X}{Z}, f'\frac{Y}{Z} \right) \tag{6}$$

where (X, Y, Z) are the coordinates of the point in the world projected in the considered pixel and f, f' are the two focal lengths before and after the zoom. This means that $x' = xf'/f$ and $y' = yf'/f$, or, written in homogeneous notation

$$\begin{bmatrix} f'/f & 0 & 0 \\ 0 & f'/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \tag{7}$$

which is an affine transform with parameters $a_{11} = a_{22} = f'/f$. Since the proposed system, as shown in section 2, is based on an affine motion model, it can successfully handle the image warping introduced by zoom. Actually more complex zoom models exist, but they show that the error introduced with the pinhole approximation is small and non noticeable if the distance of the object from the optics is much bigger than the focal length [15]. The zoom can also be actively used to improve the quality of tracking and to allow the use of high-level surveillance tasks. Murray [7] distinguishes between two purpose-driven zoom and reactive zoom and claims that the zoom can improve the performance of the system.

In our solution, the forward translation of the camera introduces a parallax movement that cannot be modelled as an affine transformation. If the camera is translated by a distance c, the point $(x, y) = (fX/Z, fY/Z)$ moves to

$$(x', y') = \left(f\frac{X}{Z-c}, f\frac{Y}{Z-c} \right) = \left(x\frac{Z}{Z-c}, y\frac{Z}{Z-c} \right) \tag{8}$$

The scaling factor $Z/(Z-c)$ is potentially different for every image point, since it depends from the distance Z of the point from the camera, but it can be approximated as constant (and consequently affine) when $Z >> c$. Since c is the translation of the camera between the acquisition of two consecutive frames, this impose an upper limit to the camera speed. If the pursued object is moving faster than the camera, then the active zoom can be used to keep the object's area constant, thus allowing a better object detection. The use of zoom also improves the alignment quality when the camera rotates in pan or tilt direction. In this case, the image transformation is not affine (it is actually projective), but can safely be approximated as an affine translation when the focal length, and consequently the zoom level, is big enough.

4 Experimental Results

The proposed method has been tested on sequences acquired in the hallways of the University building. Several experiments have been executed following a strategy that involves an incremental complexity for the tests. The sequences used for the tests have been acquired by a Cohu 3812 CCD camera mounted on a Robosoft Pioneer 2-DXe mobile indoor platform and are characterised by images of 320X240 pixels. The tests have been performed on a laptop equipped with an Athlon 2GHz processor and 256MByte of RAM. To evaluate the motion detection module a first test has been executed on sequences with no moving objects. In order to evaluate the precision of the alignment the following measure has been adopted:

$$E = \frac{\sum_i g_i \log p_i}{\sum_i \log p_i} \tag{9}$$

where g_i are the grey values and p_i the number of pixels with intensity value g_i. The value E has been computed on sequences characterised by a simple rotational movement, translational movement, longitudinal motion and by a combination of all these

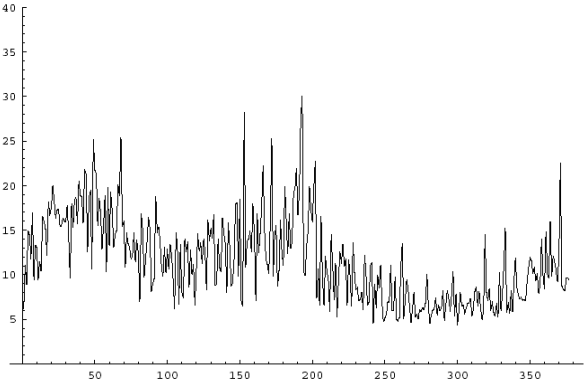
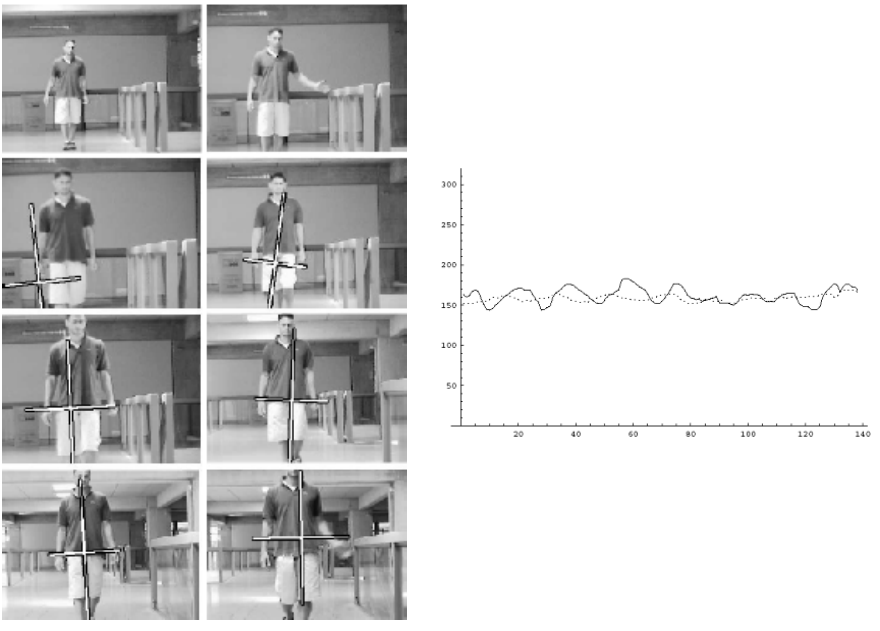


Fig. 2. Motion detection on sequences without moving objects in the scene. On the left some frames of a test sequence are shown, on the right the values of the error for the represented sequence are plotted. The mean value of the error E is 11.60.

types. The experiments significance has been increased by introducing different velocities on the mobile platform. The obtained results have pointed out a mean value for the error E equal to 12.18. Moreover, we have computed the number of iterations requested by our algorithm to converge and the average value, over all the test sequences, is equal to 1.56. Both values represent a good result by showing how the developed system performs a good alignment in few iterations. In Figure 2, the chart of the computed error for a test sequence is shown. To test the motion detection module, a second phased based on sequences presenting at least a moving object has been executed. It has been shown a good behaviour of the system as the object detection has been successful in the 97.4% of the frames. It is worth noting that the 2.6% of the cases in which the detection failed are related to spots and not to continuous frames. For these tests the average value of the error E is equal to 12.58 that is really close to the performances obtained in context of no moving objects. Regarding the number of iteration needed to converge the value 1.92 is once again very close to the previous tests value. To test the performance of the object tracking module, ten people have been asked to process the test sequences in order to define the ground truth position of the moving person barycentre. Then, the ground truth position for each frame has been computed as mean of the position marked by each person. The results obtained in context of tangent motion (objects moving orthogonally to the optical axis) showed that our method reached a good performance level. For these kinds of sequences, the mean error in the estimation of the barycentre is equal to 20.85 pixels. This error is principally due to the vertical component while if we consider only the estimation error performed on the horizontal axis the performance increases to 12.18 pixels. We can consider this a good behaviour if we analyse the motion involved in this type of sequences. Regarding the radial motion (moving object going toward to or faraway from the camera) the global error has been equal to 16.43 pixels that is lower than the error performed on the previous sequences. Also in this context, by considering only the horizontal component, the error decrease to only 9.66 pixels. Globally the module of object tracking has supplied good results allowing to maintain the object at the centre of the image, or in its closer region, for



(a) (b)

Fig. 3. Example of object tracking. On the left some frames of the sequences are shown, while in the right side the chart of the barycentre position is shown. The chart plot the horizontal position of the barycentre as marked by the testers (dotted line) and the estimated value computed by the tracking module (continuous line).

the majority of the frames belonging to the test sequences. An example of the results obtained by the proposed system can be seen in Figure 3.

5 Conclusions

In this paper, the use of an autonomous vehicle, called autonomous surveillance vehicle (ASV), for surveillance and monitoring of indoor environments has been presented. The ASV has been specifically designed for help a remote operator in the work of monitoring wide indoor areas. As it has been shown, it is able to move around a specific indoor environment (e.g., a building) and to track mobile persons. To achieve such results the system is composed by two main modules:a) object detection and b)object tracking. The former allow to detect moving object, by first aligning two consecutive frame by interleaving multiresolution and iterative methods and therefore applying a change detection technique to highlight pixels belonging to mobile objects. The latter, is based on a Kalman Filter in which the prediction phase has been modified to take into account the previous results in the detection. Several experiments on indoor sequences demonstrate that the proposed ASV is able to detect the motion inside the monitored scene, then to perform a good identification of the mobile objects. Finally, such objects can be tracked with an accuracy that allows to maintain the objects inside the field of view.

References

1. M. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications to early vision. *International Journal of Computer Vision*, 19(1):57–92, 1996.
2. I. Cohen and G. Medioni. Detecting and tracking moving objects in video from and airborne observer. In *Procceddings of the IEEE Image Understanding Workshop*, pages 217–222, Manterey,CA, November 1998.
3. R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. In *Proceedings of the IEEE*, volume 89, pages 1456–1477, October 2001.
4. G.N. DeSouza and A.C. Kak. Vision for mobile robot navigation: A survey. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 24(2):237–267, 2002.
5. G.L. Foresti and C. Micheloni. A robust feature tracker for active surveillance of outdoor scenes. *Electronic Letters on Computer Vision and Image Analysis*, 1(1):21–36, 2003.
6. G. Giralt, R. Sobek, and R. Chatila. A multi-level planning and navigation system for a mobile robot; a first approach to hilare. In *Proceedings of International Joint Conference of Artificial Intelligence*, volume 1, pages 335–337, 1979.
7. E. Hayman, T. Thorhallsson, and D. Murray. Zoom-invariant tracking using points and lines in affine views: an application of the affine multifocal tensors. In *International Conference on Computer Vision*, pages 269–277, 1999.
8. M. Herbert, C. Thorpe, and S. Stenz. *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*. Kluwer Academic, 1997.
9. B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
10. H.P. Moravec. The standford cart and the cmu rover. *Proc. IEEE*, 71(7):872–874, 1983.
11. J. Rosenblatt, S. Williams, and H.Durrant-Whyte. Behavior-based control for autonomous underwater exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco,CA, April 2000.
12. P. Rosin. Thresholding for change detection. In *Proceedings of IEEE International Conference on Computer Vision*, pages 274–279, Bombay India, 1998.
13. B. Southall, T. Hague, J.A. Marchant, and B.F. Buxton. Vision-aided outdoor navigation of an autonomous horticultural vehicle. In *Proceeding of the first International Conference on Vision Systems*, 1999.
14. G. P. Stein and A. Shashua. Model-based brightness constraints: on direct estimation of structure and motion. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 22(9):992–1015, 2000.
15. B.J. Tordoff and D.W. Murray. Reactive control of zoom while tracking using perspective and affine cameras. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 26(1):98–112, 2004.
16. P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE CVPR*, 2001.

Track Matching by Major Color Histograms Matching and Post-matching Integration

Eric Dahai Cheng and Massimo Piccardi

Faculty of Information Technology, University of Technology, Sydney (UTS),
PO Box 123, Broadway NSW 2007, Australia
{cheng, massimo}@it.uts.edu.au

Abstract. In this paper we present a track matching algorithm based on the "major color" histograms matching and the post-matching integration useful for tracking a single object across multiple, limitedly disjoint cameras. First, the Major Color Spectrum Histogram (MCSH) is introduced to represent a moving object in a single frame by its most frequent colors only. Then, a two-directional similarity measurement based on the MCHS is used to measure the similarity of any two given moving objects in single frames. Finally, our track matching algorithm extends the single-frame matching along the objects' tracks by a post-matching integration algorithm. Experimental results presented in this paper show the accuracy of the proposed track matching algorithm: the similarity of two tracks from the same moving objects has proved as high as 95%, while the similarity of two tracks from different moving objects has been kept as low as up to 28%. The post-matching integration step proves able to remove detailed errors occurring at the frame level, thus making track matching more robust and reliable.

1 Introduction

Being able to track a single object throughout a network of cameras is an important function for effective video surveillance of wide areas [1-7]. However, in most real-world camera networks it is not possible to track a moving object through a continuity of overlapping camera views. Instead, most often the object has to completely exit from the view of a certain camera before it can reappear under the view of a different one. This common scenario is often referred to as disjoint camera views, where observations of a same object are disjoint in time and space to a certain extent. In order to allow tracking in such a scenario, single-camera tracks of a same object must be matched across neighbouring cameras. The assumption in this work is that tracks are available from within single camera views, and the goal is to find correspondences between such tracks.

Accordingly, in this paper we present a track matching algorithm based on the "major color" histograms matching and the post-matching integration. First, a color distance based on a geometric distance between two points in the RGB space is introduced to measure the similarity of any two colors. By using the color distance and a given threshold, all pixels from a moving object MO_i in a given frame t are clustered into a limited number of colors, with each color's frequency defined as the number of

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1148 – 1157, 2005.

pixels with that color. Such colors are then sorted in descending frequency order and the first k used to represent the moving object. We call this histogram the major color spectrum histogram (MCSH) representation of $MO_{i,t}$. Given two arbitrary moving objects, $MO_{i,t}$ and $MO_{j,u}$ from two different frames, t and u, a similarity criterion based on the major color representation is used to assess their matching (single-frame matching). The single-frame matching is then extended along the two moving objects' tracks by selecting a same number of frames in each track, performing the matching between the corresponding frames of each track, and integrating the matching results along time. Finally, the time-integrated decision is compared against an assigned threshold to provide the final track matching decision. To the best of our knowledge, this is one of the first papers in the current literature to tackle the problem of track matching across disjoint camera views [8, 9]. Differently from those previous papers, our approach does not require global track matching [8] or rely on a topographic model of the camera network [9].

2 Major Color Spectrum Histogram

2.1 Concept of Color Distance

In this paper, we first introduce a "color distance" between two color pixels in the RGB space based on a normalized geometric distance between the two pixels. Such a geometric distance is defined in equation (1) and exemplified in Fig. 1.

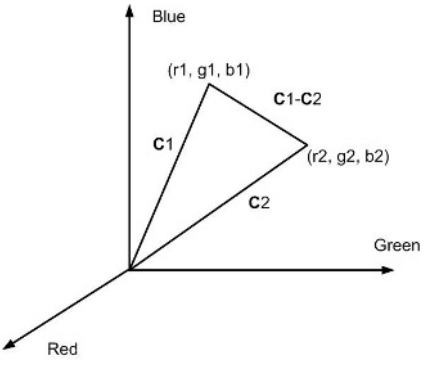


Fig. 1. The distance between 2 color pixels in RGB space

$$d(C_1, C_2) = \frac{\|C_1 - C_2\|}{\|C_1\| + \|C_2\|} = \frac{\sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}}{\sqrt{r_1^2 + g_1^2 + b_1^2} + \sqrt{r_2^2 + g_2^2 + b_2^2}} \quad (1)$$

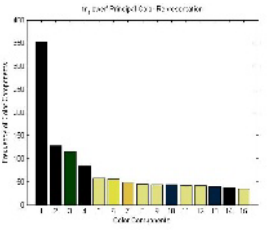
Where C_1 and C_2 are the color vectors shown in Fig. 1.

2.2 Moving Object Major Color Representation

In the RGB color space, using 1 byte to represent each color yields a total of 16.8 million (16,777,216) different colors. It is, in general, very difficult to compare two

objects based on so many possible values. By using the concept of color distance, we can scale down the possible colors from 16.8 million to a very limited number of "Major Colors" (for example, 15 to 100) without losing much accuracy on representing a moving object. For each moving object, a given number of major colors are retained in the representation, while colors that rarely appear are discarded [10,11]. Colors within a given mutual distance threshold are dealt with as a single color. An example of such a major color representation is shown in Fig. 2.

(a) 'tn_flower' picture



(b) MCR Histogram (threshold: 0.01)

Fig. 2. The Major Color Spectrum Histogram (MCSH) of 'tn_flower'

In Fig. 2 (a), the example picture ('tn_flower') is shown. In this picture, we can see that the most frequent colors are around dark green-black and yellow values. Fig. 2 (b) shows us the histogram of the major colors under the color distance threshold of 0.01. In the histogram, we can see that there are 4 main dark green-black bins with the highest frequencies (bins 1-4). The numbers of dark green-black pixels falling in these bins are about 350, 125, 120 and 85 respectively. The yellow colors are distributed in color spectrum bins 5, 6, 7, 8, 9, 11, 12 and 15. The numbers of pixels of yellow colors are between about 60 and 30. There are also 3 dark green-black bins spread in bins 10, 13 and 14, with the pixel numbers between 40 and 35.

3 Single-Frame Matching and Post-matching Integration Algorithm

3.1 Moving Objects Similarity

In this paper, the similarity measurement between two moving objects is based on their color histograms [12-14]. In particular, we use the Major Color Spectrum Histogram (MCSH) which is a generalization of [14] in that the objects do not need to be the output of a background subtraction process. We assume that there are *M* major

$$MCS(A) = \{C_{A_1}, C_{A_2}, \cdots, C_{A_M}\}. \tag{2}$$

colors in the spectrum of moving object A, which can be represented as:
Where $C_{A_i}, i = 1,2,\cdots,M$ is the color vector (RGB) of major colors in object A. Object A's color spectrum histogram (i.e. the frequencies) can be represented as:

$$p(A) = \{p(A_1), p(A_2), \cdots, p(A_M)\}. \tag{3}$$

Similarly, the major color spectrum of object B can be represented as follows:

$$MCS(B) = \{C_{B_1}, C_{B_2}, \cdots, C_{B_N}\}. \tag{4}$$

Where $C_{B_j}, j = 1,2,\cdots,N$ are the color vectors (RGB) of major colors in object B. Object B's color spectrum histogram can then be represented as:

$$p(B) = \{p(B_1), p(B_2), \cdots, p(B_N)\}. \tag{5}$$

In order to define the similarity between two moving objects, a *visibility measurement* of major colors C_{A_i} from moving object B's major color $MCS(B)$ is defined as:

$$p(A_i \mid B) = \min\{p(A_i), \sum_{C_{B_j}: d(C_{A_i}, C_{B_j}) < \sigma} p(B_j)\} \tag{6}$$

with $i = 1,2,\cdots,M$, and $j = 1,2,\cdots,N$. The above equation shows us that the visibility of A from B is given by the sum of histogram values of all major colors in moving object B that are close to the color C_{A_i} (i.e. the color distance between C_{A_i} and C_{B_j} is less than a threshold , for example, 0.01, i.e. $d(C_{Ai}, C_{Bj}) < \sigma$). The 'min' operation ensures that $\sum p(A_i \mid B) \le \sum p(A_i)$. So, the similarity of moving object B to moving object A is defined as:

$$Similarity(B \to A) = \frac{\sum_{i=1,2,\cdots,M} p(A_i \mid B)}{\sum_{i=1,2,\cdots,M} p(A_i)} \tag{7}$$

Similarly, the similarity of moving object A to moving object B is defined as:

$$Similarity(A \to B) = \frac{\sum_{j=1,2,\cdots,N} p(B_j \mid A)}{\sum_{j=1,2,\cdots,N} p(B_j)} \tag{8}$$

where $p(B_j \mid A)$ is defined as:

$$p(B_j \mid A) = \min\{p(B_j), \sum_{C_{A_i}: d(C_{A_i}, C_{B_j}) < \sigma} p(A_i)\} \quad . \tag{9}$$

If both visual objects are the same physical object, both similarities will be high (close to 1.0). Otherwise, both will be low, or at least one will be low (much lower than 1.0). So, we define the overall similarity between moving object A and moving object B as:

$$Similarity(A, B) = \min\{Simililarity(A \to B), Simililarity(B \to A)\} \tag{10}$$

3.2 Single-Frame Matching and Post-matching Integration Algorithm

In the track matching algorithm, we consider a same number of frames from each track. The algorithm is shown in Fig. 3.

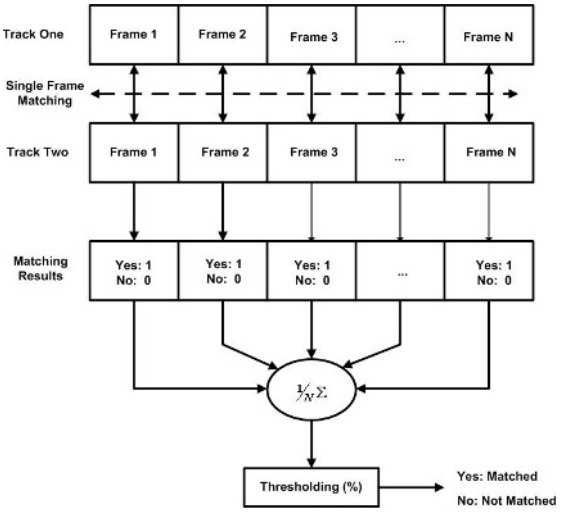


Fig. 3. Single-frame matching and post-matching integration algorithm

Fig. 3 shows us the single-frame matching as the first step of our algorithm. Moving objects from corresponding frames in Track One and Track Two are matched based on similarity of their major color spectrum, and the matching results are given as a binary decision. The second step is the multi-frame post-integration, normalization, and thresholding. The advantages of this algorithm are:

1) The single-frame matching is based on the major color spectrum histogram and two direction similarities measurements, which makes the single-frame matching very accurate.
2) The final conclusion is made based on the statistical average of single-frame matching. So, no detailed feature errors are carried forward after this stage, which makes the track matching conclusion more reliable than single frame matching.

4 Experimental Results and Analysis

In our experiments, we segment and track moving objects based on [6, 7]. In the following, we report example results from four typical tracks from the PETS 2001 dataset where three moving objects have been detected and tracked, namely a white van, a female person and a male person. The segmented moving objects, major color spectrum histograms and experimental results are shown in the following sections.

4.1 The Matching of the Same Moving Van in Two Different Tracks

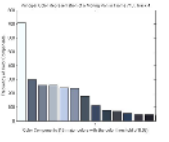
The test data here reported are from a white moving van (reference: track 4, frames 700-745, and track 5, frames 900-945.). Typical frames, the extracted moving objects, their masks and major color spectrum histograms are shown in Figs. 4 and 5. (The segmentation was done by a previous researcher in our research group, in which occlusion and shadowing were not taken into account.) The single frame matching

and post-matching integration results with the color distance threshold of 0.01, single frame matching threshold of 0.75, final integration matching threshold of 0.8 are shown in table 1.

(a) A moving van in frame 710 track 4

(b) Extracted moving object

(c) Moving object mask



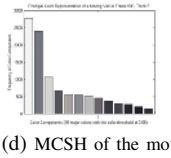
(d) MCSH of the moving van (Color threshold: 0.06)

Fig. 4. Major Color Spectrum Histogram (MCSH) of a moving van (frame 710, track 4)

(a) A moving van in frame 900 track 5

(b) Extracted moving object

(c) Moving object mask



(d) MCSH of the moving van (Color threshold: 0.06)

Fig. 5. Major Color Spectrum Histogram (MCSH) of a moving van (frame 900, track 5)

Table 1. Results of Single Frame Matching and Post-Integration (Track 4, Frame 700-745/Track 5, Frame 900-945)

Test Case	Frame Index No	Track No	No of colors	Single Frame Similarity	Matching Results(1/0)
1(700_4/900_5)	700	4	110	0.803	1 (Yes)
	900	5	601		
2(705_4/905_5)	705	4	59	0.866	1 (Yes)
	905	5	253		
3(710_4/910_5)	710	4	17	0.805	1 (Yes)
	910	5	191		
4(715_4/915_5)	715	4	29	0.726	0 (No)
	915	5	69		
5(720_4/920_5)	720	4	41	0.946	1 (Yes)
	920	5	54		
6(725_4/925_5)	725	4	22	0.800	1 (Yes)
	925	5	135		
7(730_4/930_5)	730	4	7	0.807	1 (Yes)
	930	5	135		
8(735_4/935_5)	735	4	41	0.809	1 (Yes)
	935	5	148		
9(740_4/940_5)	740	4	29	0.798	1 (Yes)
	940	5	85		
10(745_4/945_5)	745	4	20	0.880	1(Yes)
	945	5	85		
Single Frame Matching Integration	7000-745	4	N/A	N/A	0.9 (90%)
	900-945	5	N/A		

The test results in table 1 show that:

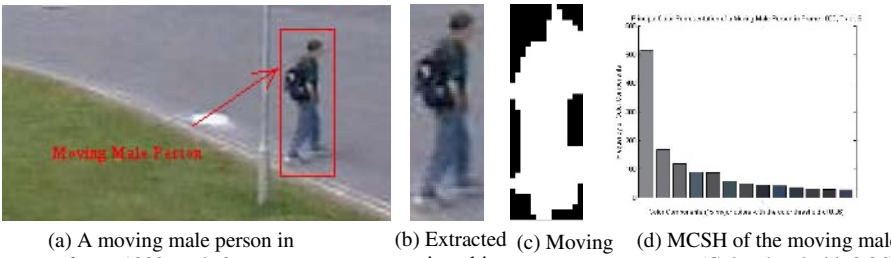
1) In most single frame matching test cases, the similarities are between 0.80 to 0.95 which are higher than the single frame matching threshold, Th_{sf} = 0.75 (75%). Thus, the moving objects in these frames are correctly matched. In single frame matching case 4, the similarity is 0.73 and thus matching failed, possibly because of tracking or moving object extraction errors.
2) In the example, the post-matching integration rate is 0.9 (90%), which is higher than the final matching threshold Th_{track} = 0.8 (80%); thus, the correct final conclusion can be made.
3) Thanks to the accuracy of this procedure, both thresholds Th_{sf} and Th_{track} can be kept relatively high so as to strongly limit false positives.

4.2 The Matching of Two Different People from Two Different Tracks

The single-frame matching and post-matching integration results for tracks from two different people (reference: moving female person, track 2, frames 460-505; moving male person, track 6, frames 1000-1045) are shown in table 2. Typical frames, the extracted moving objects, their masks and major color spectrum histograms are shown in Figs. 6 and 7.

(a) A moving female person in frame 460 track 2 (b) Extracted moving object (c) Moving object mask (d) MCSH of the moving female person (Color threshold: 0.06)

Fig. 6. MCSH of a moving person (frame 460, track 2)



(a) A moving male person in frame 1000 track 6 (b) Extracted moving object (c) Moving object mask (d) MCSH of the moving male person (Color threshold: 0.06)

Fig. 7. MCSH of a different moving person (frame 1000, track 6)

The test results in table 2 show that:

1) The two moving objects are of similar, small size in their respective frames and their color appearances show some resemblance. However, their MCSH differ significantly. In all test cases, their similarities are between 0 and 0.278 which are well below Th_{sf}, thus they are correctly discriminated.
2) After post-matching integration, the matching rate is still 0 (0%). Therefore, the correct final decision can be easily made.
3) Even with the very limited number of major colors used (15), the correct matching was still easily achieved.

The cases described are exemplary of the accuracy of the proposed track matching algorithm. Accuracy is high when views of a same object do not change significantly, while major illumination and pose changes will eventually compromise the effectiveness of matching. In order to make application more general, we are currently developing a compensation algorithm for global illumination variations based on color calibration and background models, and an *incremental* major color spectrum histogram (IMCSH) able to cope with small pose and appearance changes occurring along the track. In addition, the track matching procedure will be eventually integrated with other geometric features such as gait-filtered height.

Table 2. Results of Single Frame Matching and Post-Integration (Number of Colors: 15; Color Threshold: 0.01; Track 2, Frame 460-505/Track 6, Frame 1000-1045)

Test Case	Frame Index No	Track No	Single Frame Similarity	Matching Results(1/0)
1 (460_2/1000_6)	460	2	0.0833	0 (No)
	1000	6		
2 (465_2/1005_6)	465	2	0.0372	0 (No)
	1005	6		
3 (470_2/1010_6)	470	2	0.0272	0 (No)
	1010	6		
4 (475_2/1015_6)	475	2	0.2197	0 (No)
	1015	6		
5 (480_2/1020_6)	480	2	0.2779	0 (No)
	1020	6		
6 (485_2/1025_6)	485	2	0.1449	0 (No)
	1025	6		
7 (490_2/1030_6)	490	2	0	0 (No)
	1030	6		
8 (495_2/1035_6)	495	2	0.1912	0 (No)
	1035	6		
9 (500_2/1040_6)	500	2	0.0281	0 (No)
	1040	6		
10 (505_2/1045_6)	505	2	0.0394	0 (No)
	1045	6		
Single Frame Matching Integration	460-505	2	N/A	0 (0%)
	1000-1045	6		

5 Conclusions

In this paper, a track matching algorithm has been proposed to match tracks from single objects across non-overlapping camera views. First, a color distance based on a normalized geometric distance between two points in the RGB space is defined and used to measure similarity of two different colors. Then, a Major Color Spectrum Histogram is introduced to represent a moving object by its "major colors" and their frequencies. A similarity measurement is then used to measure the similarity of any two moving objects. Finally, track matching is based on the post-matching integration of single-frame matching.

Based on our experimental results, the following conclusions can be drawn:

1) Experimental results shown that the major color spectrum histogram (MCSH) based on the given color distance proved able to represent moving objects accurately with a limited number of colors and their frequencies.
2) In the experiments reported, the similarity of a same moving object in two different tracks has reached as high as 95%, while the similarity of two different moving objects has been kept as low as 0% to 28%. This allowed us to use a relatively high threshold (75%) able to limit false positive errors.
3) Since the post-matching integration is based on single-frame matching binary results, no detailed feature error is carried forward after this stage. Moreover, post-matching integration makes track matching more robust and reliable than single frame matching.

The proposed track matching algorithm can significantly extend current video surveillance applications by providing them with accurate tracking across non-overlapping camera views which is the actual case for many real-world surveillance camera networks.

Acknowledgment

This research is supported by the Australian Research Council, ARC Discovery Grant Scheme 2004 (DP0452657).

References

1. T.H. Chang and S. Gong, "Tracking Multiple People with a Multi-Camera System", Proceedings of the 2001 IEEE Workshop on Multi-Object Tracking, 19-26, 2001.
2. I. Haritaoglu, D. Harwood and L.S. Davis, "W4 Real-Time Surveillance of People and Their Activities", IEEE Trans. on PAMI., 22(8), 809-830, 2000.
3. L.M. Fuentes and S.A. Velastin, "People Tracking in Surveillance Applications", Proceedings of the 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance, 2001.
4. C. Wren, A. Azarbaygaui, T. Darrell and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," IEEE Trans. PAMI, 19(7), 780-785, 1997.
5. A. Lipton, H. Fujiyoshi, and R. Patil, "Moving target classification and tracking from real-time video," in Proc. Of the IEEE Image Understanding Workshop, 1998, pp. 129-136.

6. A. Elgammal, D. Harwood, and L. Davis. “Non-parametric Model for Background Subtraction”, 6th European Conference on Computer Vision, 2000.
7. A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle, “Appearance Models for Occlusion Handling”, 2nd Int. Workshop on Performance Evaluation of Tracking and Surveillance Systems, 2001.
8. O. Kaved, Z. Rasheed, K. Shafique, M. Shah, “Tracking Across Multiple Cameras With Disjoint Views,” in Proc. of the Ninth IEEE Int. Conf. on Computer Vision (ICCV’03), vol. 2, pp. 952-957.
9. D. Makris, T. Ellis, and J. Black, “Bridging the Gaps between Cameras,” in Proc. of the 2004 IEEE CS Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 205-210.
10. Zoran Zivkovic and Ben Krose, “An EM-like algorithm for color-histogram-based object tracking,” in Proc. IEEE CVPR 2004.
11. W. Lu and Y. P. Tan, “A Color Histogram Based People Tracking System”, Proc. IEEE Int’l Symp. Circuits and Systems, vol. 2, pp. 137-140, 2001.
12. Y. Chen and E. Wong, “Augmented image histogram for image and video similarity search,” in Proc. SPIE Storage and Retrieval for Image and Video Databases,” pp. 523-532.
13. J. Hu and A. Mojsolovic, “Optimum color composition matching of images,” in Proc. 15th Int. Conf. on Pattern Recognition,” vol. 4, pp. 47-51, Barcelona, Spain, 2000.
14. Liyuan Li, Weimin Huang, I.Y.H. Gu, K. Leman, Qi Tian, “Principal Color Representation for Tracking Persons,” in Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics 2003, vol. 1, pp. 1007-1012.

Robust Particle Filtering for Object Tracking

Daniel Rowe, Ignasi Rius, Jordi Gonzàlez, and Juan J. Villanueva

Computer Vision Centre/Department of Computer Science,
Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain
drowe@cvc.uab.es

Abstract. This paper addresses the filtering problem when no assumption about linearity or gaussianity is made on the involved density functions. This approach, widely known as *particle filtering*, has been explored by several previous algorithms, including *Condensation*. Although it represented a new paradigm and promising results have been achieved, it has several unpleasant behaviours. We highlight these misbehaviours and propose an algorithm which deals with them. A test-bed, which allows proof-testing of new approaches, has been developed. The proposal has been successfully tested using both synthetic and real sequences.

1 Introduction

The increasing interest in tracking is motivated by the huge number of promising applications that can now be tackled in real time thanks to recent technological advances. These applications include performance analysis of human motion, surveillance, video-indexing and smart interfaces, among others.

However, tracking can be extremely complex and time-consuming. In the first place, strong requirements are mandatory. Real-time processing, extreme robust performances or high accuracy may be critical. Moreover, several difficulties must be expected such as multiple-target tracking with unknown and highly non-linear dynamics, presence of heavy background clutter, strong noise and partial occlusions.

This paper focuses on tracking by means of particle-filtering in the conditions described above. This approach has been explored by several previous algorithms, including *Condensation* [3]. Although some results were achieved, *Condensation*-based algorithms have several unpleasant behaviours. In this paper, we highlight these misbehaviours and propose an algorithm which deals with them, also in presence of the above-presented tracking difficulties.

The remainder of this paper is organized as follows. Section 2 covers the probabilistic framework and revises *Condensatio*n, exposing its misbehaviours. Section 3 describes the proposed algorithm. Section 4 shows experimental results with both synthetic and real sequences. Section 5 concludes this paper.

2 Particle-Filtering Background

A probabilistic framework is commonly used as a way to perform tracking [5]. Classical approaches, such as the *Kalman Filter*, rely on linearity and gaussian-

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1158–1165, 2005.

ity assumptions about the involved distributions. More recent works make use of *Bayesian filters* combined with *Monte Carlo Simulation* methods in order to deal with nonlinear and non-Gaussian transition and sensor models [1,2]. However, these methods present several important drawbacks. A great number of improvements have been introduced in recent years [7,6] but there is still much ground to cover before solving unconstrained tracking.

We are interested in *filtering*, that is, the computation of the belief state $\mathbf{S}_t$ given all evidence to date $\mathbf{e}_{1:t}$. Posteriors are calculated through *recursive estimation*:

$$P(\mathbf{S}_t \mid \mathbf{e}_{1:t}) = \underbrace{P(\mathbf{e}_t \mid \mathbf{S}_t)}_{\text{likelihood}} \int \underbrace{P(\mathbf{S}_t \mid \mathbf{s}_{t-1})}_{\text{transition mod.}} \underbrace{p(\mathbf{s}_{t-1} \mid \mathbf{e}_{1:t-1})}_{\text{previous post.}} d\mathbf{s}_{t-1}. \tag{1}$$

updating prediction stage

This pdf is projected forward according to the transition model, making a prediction, and it is updated in agreement with the likelihood function value based on the new evidence.

Unfortunately, recursive estimation leads to expressions that are impossible to evaluate analytically unless strong assumptions are applied. This problem is overcome by simulating N i.i.d. samples which approximate the posterior pdf. This approach is known as *particle filtering* within the control field and *survival of the fittest* in artificial intelligence. Such methods, renamed as *Condensation* [3], were first introduced in the computer vision research area by Isard and Blake.

The method works as follows: the posterior pdf at time $t-1$ is represented by a set of particles or samples, $\left\{\mathbf{s}_{t-1}^i; i = 1:N\right\}$. The temporal prior $\left\{\hat{\mathbf{s}}_t^i\right\}$ is obtained by applying the transition model $P(\mathbf{S}_t \mid \mathbf{s}_{t-1})$ to each sample. The likelihood $P(\mathbf{e}_t \mid \mathbf{S}_t)$ is represented by weights π_t^i according to the likelihood values at the sample points. Once all samples have been propagated and measured, the set is re-sampled using normalized weights $\overline{\pi}_t^i$ as probabilities. The sample set $\left\{\mathbf{s}_t^i; i = 1:N\right\}$ represents the posterior at time t. Expectations can be approximated as:

$$\mathbb{E}_{P(\mathbf{S}_t|e_{1:t})}(\mathbf{S}_t) \simeq \sum_{i=1}^N \overline{\pi}_t^i \hat{\mathbf{s}}_t^i \simeq \frac{1}{N} \sum_{i=1}^N \mathbf{s}_t^i. \tag{2}$$

However, *Condensation* has several unpleasant behaviours as stated in [4]. *Sampling impoverishment* is one of the main drawbacks of re-sampling algorithms. Samples are spread around several modes indicating hypotheses in the state space. Nevertheless, some of them are spurious. Similarly to genetic drift, there is a non-negligible probability of losing modes, a low probability of recovering them and the remaining modes could be all spurious. It can also be derived from this fact that different runs of the algorithm may lead to different results. Therefore, computed expectations in different runs have high variance although computed expectations within the same algorithm run have low variance, making the tracker look stable. In addition, the sample set size N is kept constant over

time and there is no information about how large N should be for a requested precision. Once N has been heuristically set, it may happen that at later times larger values of N may be required. Finally, *Condensation* was designed to keep multiple hypotheses, but only for a single target.

3 Towards Robust Tracking

We propose an algorithm based on particle filtering. In contrast with *Condensation*, in which the target is represented by means of contours, we rely on a pixel-wise approach since it is expected to be more robust to partial occlusions.

The motion of the centre of a bounding box is modelled using first-order dynamics in image coordinates. The l-labeled target's state is defined as $\mathbf{s}_t^l = \left(\mathbf{x}_t^l, \mathbf{u}_t^l, \mathbf{w}_t^l, \mathbf{A}_t^l\right)^T$. Each component represents the position, speed, bounding-box size and appearance, respectively. A label associates one specific appearance model to the corresponding samples, allowing multiple-target tracking.

3.1 Likelihood Function

The likelihood function gives the pdf of image features given the state. The selected features are pixel-oriented. Thus, the appearance will be given by a matrix whose elements are the pixel intensity values. Therefore, it can be assumed that the appearance is independent from the speed component. Let $\mathbf{I}_t$ be a matrix whose elements are the scene pixel intensity values at time t. Given the predicted position $\mathbf{x}_t = (x, y)$ and bounding-box size $\mathbf{w}_t = (w, h)$, the corresponding image subregion is denoted by $\mathbf{I}_t^p$. The model appearance matrix must be scaled according to the sample size. Let $\mathbf{A}_t^s$ be the scaled matrix for the model. Considering a smooth process, we assume that the appearance is constant between frames. Assuming also *White Additive Gaussian Noise* (WAGN), the likelihood function can be expressed as:

$$\begin{aligned} P\left(\mathbf{I}_t \mid \mathbf{S}_t\right) &= P\left(\mathbf{I}_t^p \mid \mathbf{A}_t^s\right) \\ &= \tfrac{1}{M} \sum_{a,b \in \mathbf{A}_t^s} \mathcal{N}\left(\mathbf{I}_t^p\left(a,b\right); \mathbf{A}_t^s\left(a,b\right), \sigma_a^2\right). \end{aligned} \tag{3}$$

where M is the number of pixels of the appearance model, (a, b) defines a pixel position in the appearance matrix $\mathbf{A}_t^s$ and σ_a^2 is the estimated camera noise variance, which randomly influences the pixel intensity values.

3.2 Weight Normalization

Multiple-target tracking causes several problems including that the target with higher likelihood may monopolize the sample set. If data association is feasible, using a prior density to generate new samples reduces the risk of sampling impoverishment. However, this is not always possible, specially during occlusions or while the target is surrounded by background clutter. In this case, those targets whose samples exhibit lower likelihood have higher probability of being lost,

since the probability of propagating one mode is proportional to the cumulative weights of the samples that constitute it.

Two kind of modes can be distinguished: samples with different labels belong to different modes, and thereby, several targets can be tracked simultaneously; secondly, samples with the same label could be spread around different modes. This fact allows us to keep several hypotheses for a single target; hopefully, one of them represents the true target state and the others are due to background clutter. In order to avoid single-target modes absorbing other target samples, *genetic drift* must be prevented. This fact happens due to the lack of *genetic memory*: we propose to include a memory term which takes into account the number of targets being tracked. Hence, weights are normalized according to:

$$\overline{\pi}_t^{i,l} = \frac{\pi_t^{i,l}}{\sum_{i=1}^{N} \pi_t^{i,j}} \frac{1}{L}, \quad \text{where } j = l, \tag{4}$$

where L is the number of targets being tracked. It does not assign a fixed number of samples to each target but ensures that each target will have the same probability of being propagated.

On the other hand, modes due to clutter are pruned because of differences in their dynamics. It is unlikely that any sample tracks local clutter since it implies highly abrupt changes in the dynamics. Thus, non-loss of the true mode depends on how the different hypotheses are generated.

3.3 Sample Dynamics

Since our likelihood function depends on the sample position but does not depend on its speed, propagated samples could have a small position error, but their speed values could become completely different from the true one in a few frames. Targets could be tracked since we are in a multiple-hypothesis scenario, but an important proportion of samples will be wasted.

Thus, we feed back the estimated target speed at time $t-1$, denoted as $\mathbf{u}_{t-1}^{l}$, into the prediction. Subsequently, speeds are not estimated from the selected samples, but calculated according to the history of positions. Both position and speed estimation are regularised according to their corresponding histories. Thus, predictions are computed as follows:

$$\hat{\mathbf{x}}_t^{i,l} = \mathbf{x}_{t-1}^{i,l} + \mathbf{u}_{t-1}^{i,l} \Delta_t + \xi_{\mathbf{x}}^{i,l}, \qquad \hat{\mathbf{u}}_t^{i,l} = \mathbf{u}_{t-1}^{l} + \xi_{\mathbf{u}}^{i,l}. \tag{5}$$

The random terms $\xi_{\mathbf{x}}^{i,l}, \xi_{\mathbf{u}}^{i,l}$ provide the system with a diversity of hypotheses and, subsequently, samples with high likelihood are more probably propagated. The l-target position and speed are estimated according to:

$$\begin{aligned} \mathbf{x}_t^{l} &= \left(\mathbf{x}_{t-1}^{l} + \mathbf{u}_{t-1}^{l} \Delta_t\right)(1 - \alpha_{\mathbf{x}}) + \left(\frac{1}{N_l} \sum_{i=1}^{N} \overline{\pi}_t^{i,l} \hat{\mathbf{x}}_t^{i,l}\right) \alpha_{\mathbf{x}}, \\ \mathbf{u}_t^{l} &= \mathbf{u}_{t-1}^{l}(1 - \alpha_{\mathbf{u}}) + \left(\mathbf{x}_t^{l} - \mathbf{x}_{t-1}^{l}\right) \alpha_{\mathbf{u}}, \end{aligned} \tag{6}$$

where $\alpha_{\mathbf{x}}, \alpha_{\mathbf{u}}$ denote the adaptation rates and N_l the number of samples of the l-target. The speed is fed back when predicting the following sample state.

3.4 Sample Size Propagation

The target dimensions are predicted using a *random-walk* model:

$$\hat{\mathbf{w}}_t^{i,l} = \mathbf{w}_{t-1}^{i,l} + \xi_{\mathbf{w}}^{i,l}, \tag{7}$$

where $\xi_{\mathbf{w}}^{i,l}$ is a random term obtained according to the target's size covariance. Sample likelihoods are calculated as the mean of the probabilities of belonging to the target of each pixel within the bounding box. Thus, it depends on the normalised overlapping area A_o between the target's bounding-box and a misaligned sample one, given by:

$$A_o = 1 - \frac{1}{w}\Delta x - \frac{1}{h}\Delta y + \frac{\Delta x * \Delta y}{w * h}. \tag{8}$$

Thus, the effects of misalignments depend on the target's size: bigger targets tolerate bigger shifts without significant likelihood falls. We are interested in the relation between the likelihoods of two particles since, once all weights are normalised, the probability of choosing one particle instead of other depends on this relation. This relation must make the 'aligned' sample likely enough to be re-sampled instead of the misaligned one. This is achieved by mapping sample likelihoods using monomial functions. Exponents are selected on-line, according to the target size, ensuring a minimum slope of the overlapping area.

On the other hand, those samples with smaller sizes are expected to have higher likelihoods since a likelihood computation involves a normalization according to the sample size and smaller samples can have relative bigger overlapping regions. As it is empirically proved, samples tend to smaller sizes and, eventually, collapse. These results strongly suggest that higher bounding boxes should be slightly favoured in the re-sampling process. Therefore, a term related to the relative sample sizes is included in the likelihood computation.

4 Experimental Results

The performance of the algorithm has been tested using both synthetic and real data. Synthetic data allow us to achieve two goals: we have access to the ground truth, therefore deviations and performance can be accurately measured;

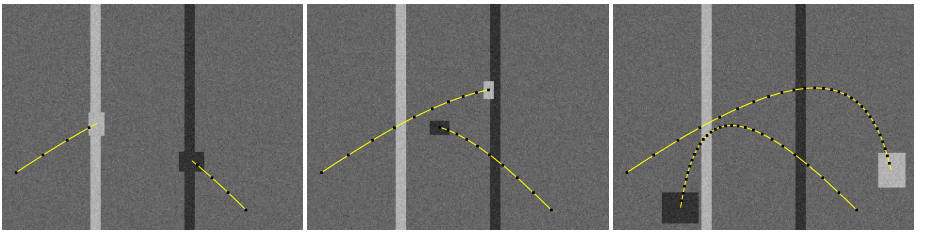
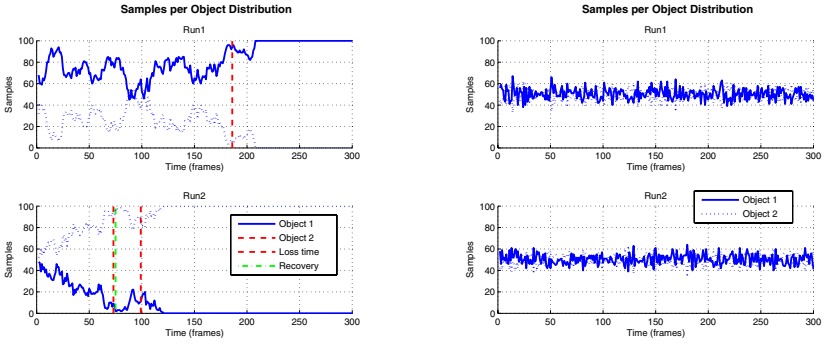


Fig. 1. Ground Truth

in addition, experimental conditions can get harder and harder on each aspect independently, thereby maximum performances can be measured.

A two-moving-target synthetic experiment has been designed. It covers several difficulties a tracker can run into, see Fig. 1. The background pixel intensity values follow a normal distribution. Both targets' pixel intensity values also have a normal distribution around different means. Two vertical strips are drawn in the background, simulating heavy clutter. Their distributions are identical to those of both targets, thereby mimicking them. Strong acquisition-device noise, modelled as WAGN, is simulated. A highly non-linear dynamic is considered: both targets move as projectiles which are shot into an environment with gravity and air friction. Each target constantly reduces its width and height during the first third of the experiment, and increases them during the rest of it. Thus, areas fluctuate from 209 to 1435 pixels and aspect ratios from 1.7 to 1.3. Tracking is performed over $T = 300$ frames using $N = 100$ samples.

Without the proposed weight normalization, the tracker loses a target due to the lack of samples, see Fig. 2.(a). On the other hand, after the proposed weight normalization, the mean number of samples per target in each run fluctuates between 49.5 % and 50.5%, see Fig. 2.(b).



(a) Without proposed weigh normalisation (b) After applying the normalisation

Fig. 2. Samples per object distribution

Table 1. Mean position error

Mean normalized error		
	Target 1	Target 2
Run 1	0.1163	0.1309
Run 2	3.8864	0.1182
Run 3	0.1222	0.1226
Run 4	0.1101	2.4679

(a) Without the speed feed-back

Mean normalized error		
	Target 1	Target 2
Run 1	0.0715	0.0716
Run 2	0.0849	0.1163
Run 3	0.0987	0.1289
Run 4	0.0645	0.0595

(b) After feeding the speed back

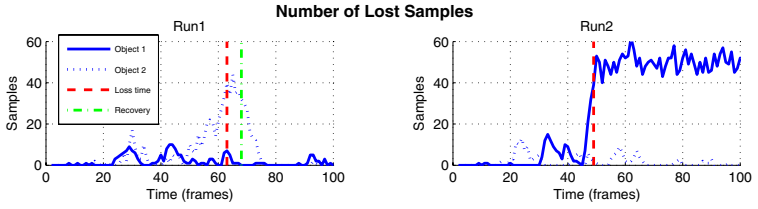


Fig. 3. Evolution of lost samples without feeding the speed back

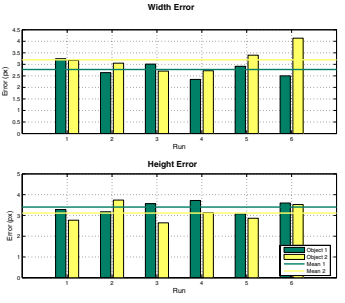


Fig. 4. Mean size error

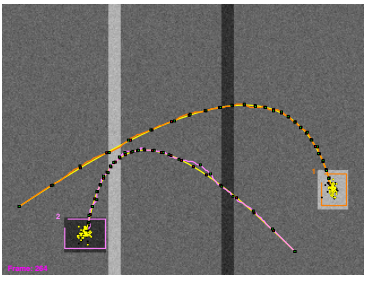


Fig. 5. Target performance

(a) (b)

Fig. 6. Target performance on real sequences

Table 1.(a) shows the mean normalized error, according to the target size, in the estimation of the target position without regularising and feeding the speed back, whereas Table 1.(b) shows the same results after applying it. A significant error reduction can be appreciated besides the fact that now none of the targets is ever lost. Fig. 3 shows the number of samples that had lost their target without our proposed speed feed-back and estimation. After applying it, the number of lost samples is negligible. The error in estimating the target size is shown in Fig. 4. Six runs for both targets are represented.

The tracker performance is shown in Figs. 5, 6: Fig. 5 presents one synthetic run; Fig. 6.(a) shows a traffic sequence; Fig. 6.(b) exhibits a traffic sequence under heavy shadow and reflectance conditions.

5 Conclusions

We have proposed a particle-filter based algorithm. It deals with multiple-target tracking whose dynamics are highly non-linear. They move through an environment with complex clutter, which mimics the target appearances, and strong noise. A new approach is taken to deal with one of *Condensation*'s great misbehaviours, the sampling impoverishment. This problem becomes critical in a multiple target-tracking scenario. The proposed sample-weight normalization prevents from the loss of any of the targets due to the lack of samples. The dynamics updating is set by feeding the estimated speed back into the prediction stage. The speed is not estimated according to the sample speeds since the likelihood does not provide speed measures. Instead, it is estimated from successive position estimations. Subsequently, both target's position and speed are regularised. Sample wastage is significantly reduced. The tracker has been successfully tested in both synthetic and real experiments concerning traffic surveillance. They are currently being applied in real applications relative to people tracking. Encouraging results are being achieved. Future research will be focused on colour-based likelihoods.

Acknowledgments. This work has been supported by the Spanish CICYT TIC 2003-08865 and the Generalitat de Catalunya Research Department (DURSI).

References

1. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *Tran. on Signal Processing*, 50(2):174–188, 2002.
2. A. Doucet. On sequential simulation-based methods for bayesian filtering. Technical Report TR310, Cambridge University, 1998.
3. M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
4. O. King and D. A. Forsyth. How does CONDENSATION behave with a finite number of samples? In *6th ECCV, Dublin, Ireland*, volume 1, pages 695–709, 2000.
5. R. Russell and P. Norvig. *Artificial Intelligence, a Modern Approach*, chapter 13-15. Prentice Hall, second edition, 2003.
6. R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan. The Unscented Particle Filter. Technical Report TR380, Cambridge University, 2000.
7. X. Varona, J. Gonzàlez, X. Roca, and J.J. Villanueva. iTrack: Image-based Probabilistic Tracking of People. In *15th ICPR, Barcelona, Spain*, volume 3, pages 1110–1113, 2000.

A Correlation-Based Approach to Recognition and Localization of the Preceding Vehicle in Highway Environments*

A. Broggi, P. Cerri, and S. Ghidoni

Dipartimento di Ingegneria dell'Informazione,
Università di Parma, Parma, I-43100, Italy

Abstract. In this paper a new approach to the problem of recognizing the preceding vehicle on highways is presented. The system is based on monocular vision. Since on highways the position of the preceding vehicle in the image varies slowly, its previous and current positions are compared using correlation. Such image processing produces a very clear output, which, at a higher level, allows a simple and fast recognition.

1 Introduction

Highways represent a particular environment for drivers: the environment is more structured than a common road, some obstacles, like pedestrians and cyclists, are not present, nor exist crossroads or traffic lights, and all vehicles move in the same direction. On the other hand, cars run very fast; this fact requires a different driving style: it is necessary to look even at very distant vehicles and understand whether they are slowing down. The most important vehicle a driver should consider is the preceding one, to which a security distance should be kept. For these reasons, an automatic system capable of understanding where the preceding vehicle is would be very useful. Such systems are already available on some cars, with the name of ACC (Adaptive Cruise Control); most of them are based on radar or infrared technology [1]. In this paper, a system based on vision performing the same task is presented. The information source is a simple low-cost grey-scale 8-bit camera, working at 10 frames per second, connected to a personal computer. The camera is installed on the top of the windscreen in a central position, and acquires 640×480 pixels images.

Many methods were developed in order to solve the discussed problem, some of them using a single camera, like the proposed one. A lot of approaches are model-based [2], [3]: even if those methods reach good results they need models that match different vehicle types. Other solutions are based on vehicle features search; simmetry is the most used feature for vehicle detection [4], [5]. The

* The work described in this paper has been developed in the framework of the Integrated Project APALACI - PReVENT, a European research activity funded by the European Commission to contribute to road safety by developing and demonstrating preventive safety applications and technologies.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1166–1173, 2005.

proposed algorithm uses a different feature: preceding vehicles are supposed to have a "U" shape[1], so the presence of two vertical edges in corrispondence to a horizontal edge is searched; other works are developed using this assumption [6]. Correlation is a well-known method for vehicle tracking [7]: in the proposed work it is used both for detection and tracking; the tracking method is similar to the method proposed by Betke [8], while just basic research has been conducted on detection using correlation.

The purpose of the system being presented is to recognize if there is a preceding vehicle going approximately at the same speed of the experimental vehicle. This means the target vehicle can be found in similar positions in two subsequent frames.

The algorithm can be divided into two parts: the first one is image processing, the second deals with target recognition and tracking. The paper reflects this organization: in Sect. 2 a description of low-level tasks is given, while Sect. 3 describes how it is possible to find a vehicle and to track it. Finally, in Sect. 4 some results are given.

2 Computation of Correlation Images

The first step of the algorithm concerns low-level tasks. Some processing is required to obtain images which can make target recognition and tracking easier. As said, these targets remain approximately in the same position within two subsequent frames; for this reason, the concept of correlation can be exploited.

2.1 Correlation of Images

Correlation is used to measure how a given quantity changes. Applying this concept to a video, correlation can produce an image that shows how objects change their position in the scene. Such an image can be produced by evaluating, for each pixel, the difference between the current image and the previous ones. The number N of considered frames of the past should therefore be defined. The grey-level of the generic pixel found in row i and column j, $P^k_{i,j}$, of the correlation image at frame k can be computed as follows:

$$P^k_{i,j} = \sum_{m=1}^{N} |G^k_{i,j} - G^{k-m}_{i,j}| \ , \tag{1}$$

where $G^k_{i,j}$ is the grey-level of the pixel in position (i, j) at frame k. The camera acquires 256 grey-level images, therefore $G^k_{i,j}$ is a number varying in the range 0–255, and the condition $P^k_{i,j} \leq N \cdot 255$ holds. This means values obtained from (1) exceed 255, and it is thus necessary to limit them by truncation $P^k_{i,j}$ to a maximum value of 255. Such an operation causes a loss of information beyond a certain variation limit, emphasizing small variations.

[1] Namely, two vertical edges and one bottom edge.

(a) (b)

Fig. 1. Examples of correlation images: correlation with 2 (a) and 19 (b) frames.

Computation of the correlation image is a time-consuming task, mainly if it involves a lot of past frames. Short computation times can be obtained reducing the use of correlation. Because of the fact that the developed algorithm is intended to work on highways, which are almost straight, it is possible to assume that preceding vehicles will be positioned in a small portion of the image. Therefore it will be sufficient to focus on this small sub-image, called window of interest, to which the evaluation of correlation can be restricted. Two examples of correlation sub-images obtained applying (1) are shown in Fig. 1. The sub-image size is 110×100 pixels, and it is overlapped to the analyzed frame. Indeed, the number of frames considered in the computation affects the white trace of the moving car. White color means that analized frames are different, while black means they are equal or, at least, similar. Figure 1 shows correlation images obtained using few (a) or many (b) past frames. After some tests, it was found that with $N = 7$ the best results were obtained.

2.2 Correlation of Edges

The concept of correlation can be further exploited by applying it to edges. A useful technique to detect vehicles is to find edges in the image using the well-known Sobel algorithm. This technique, however, also produces some noise, because lots of edges are present, due to the background and to signs painted on the road. The developed algorithm should focus on vehicles which remain almost in the same position between different frames: thus it is possible to find targets by analyzing edges that are persistent. Results of Sobel processing are therefore binarized and saved in memory, and at each new frame edges are compared to those found in the past. Finally, the algorithm generates a binarized image containing edges present at least in five of the seven frames in memory. Such image is called the edges correlation sub-image. This processing is applied only to the window of interest.

The results of the described processing are a good starting point for further steps because a lot of noise is filtered out. Examples can be found in Fig. 2, where the result is superimposed to the original frame acquired by the camera. To compare the effects of the two image processing techniques, also the correlation sub-image is reported in the frames, just under the edges correlation sub-image

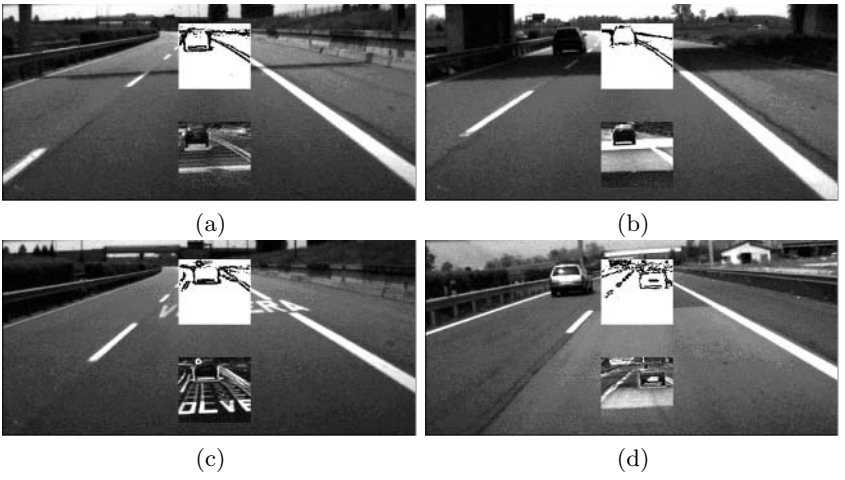


Fig. 2. Examples of low-level processing. The binarized edges correlation sub-image is superimposed on the frame in the position of the window of interest. The correlation sub-image is also shown below. In (a) and (b) thin and thick shadows do not leave any trace in the edges sub-image. In (c) a horizontal road sign is completely filtered, and in (d) the same happens to the junction of two patches of asphalt with different colors.

for a better comparison. As Fig. 2 shows, horizontal road signs and shadows are not present in the edges correlation image.

3 Vehicle Recognition and Tracking

This section deals with the problem of recognizing the preceding vehicle by analyzing the results of image processing previously described. At this level, the system switches between two different states: vehicle recognition and tracking. The algorithm starts in the recognition phase, and tries to find a vehicle in the window of interest. When this happens, it starts tracking it, and continues until it is lost; in this case, the system goes back to the recognition phase.

3.1 Recognition Phase

The aim of this phase is to test if there is a vehicle in the window of interest. The algorithm starts by analyzing the edges correlation sub-image, looking for a rectangular shape of appropriate size. From Fig. 2 it is clear that a vehicle's outline will result in long black lines in the edges correlation image; so the algorithm looks for long black edges. Because the analyzed image is binarized, as said in Sect 2.2, it is possible to enumerate black pixels for each row and column. This simple computation, however, does not consider if pixels are contiguous or not, so the same value would result for two lines with the same number of black pixel disposed in very different ways. Rather, a vehicle's edge will be an uninterrupted segment, so it is necessary to distinguish among black pixels which are contiguous from those which are not.

Edge Detection. In order to detect a vehicle's edge, a value, depending on the number of black pixels and on their placement according to a simple rule, is assigned to each row and column of the edges correlation sub-image. A weight is assigned to each black pixel, that is equal to the number of preceding black pixels plus one. Each white pixel has zero weight. Finally, all weights are added together to obtain the final number assigned to the row or column.

The described method is able to find the most visible edges. The algorithm then selects the two columns associated with the highest values to detect lateral edges of the vehicle. As an additional constraint, the two columns should be at a minimum distance from each other of five pixels, otherwise they may belong to the same edge (which may be composed by few black columns). Subsequently, the lower bound of the vehicle is searched for; this is done in the same way, but examining rows instead of columns. Because the left and right edges are already known, a rough estimation of the vehicle's size in the image can be done, allowing a restriction of the lower edge search area. Often the upper edge of the vehicle is not present in the region where the correlation is computed; in other cases, it is positioned in a region where the background produces a lot of noise, as can be observed in Fig. 2 (a), (c) and (d). Thus, the algorithm does not look for the vehicle roof, which however is not important for the estimation of the vehicle's position, and can therefore be neglected: the roof position can be used to classify the kind of vehicle. A fictious upper edge is nevertheless chosen to represent the upper limit of the window of interest, because some checks should be anyway carried out on the area inside the edges. Even if the upper limit does not come from a real detection, when a vehicle is in the scene, it is displayed in a rectangle. In this way a box that may contain a vehicle is anyway obtained.

Vehicle Check. Edges computed in the way described so far are found in each frame. It is therefore necessary to understand whether or not a vehicle is present, so some checks are performed. If at least one fails, no vehicle is found in the current frame, otherwise the recognition is successful.

Size Check. The first test is simple and concerns the mutual positions of the lateral and lower edges of the vehicle. Moreover, if an edge is shorter than a threshold (ten pixels), the check fails, because the recognition of a vehicle which is too small in the image would be too unreliable.

Correlation Check. The second test considers the correlation sub-image, and verifies that in the rectangle of interest the correlation is high: this ensures that the box represents a vehicle preceding the experimental one, going at about the same speed. This test is carried out by adding together all values of the correlation image in the rectangle of interest, normalizing it over the total number of pixels and comparing the obtained value to a threshold.

Edge Check. The last check is performed on the edge correlation sub-image. As said, the selected edges are the row and columns with the highest numbers of black pixels, considered together with their positions. But if the sub-image

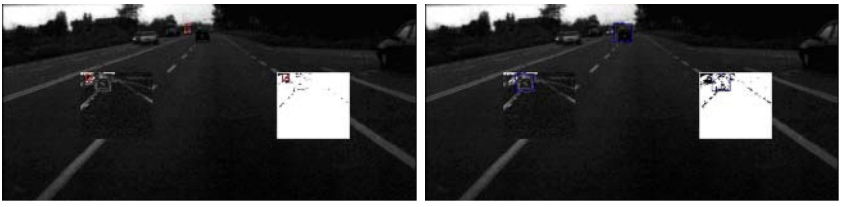
has very few black pixels, edges will be composed by a large number of white pixels, and the recognition will be too unreliable. To avoid this situation, the third check verifies that the edges are composed by pixels that are black for a fraction exceeding a threshold. The fraction that gave best results is 40%: it could seem low, but it is enough to filter out the great majority of false positives, still keeping a good recognition. This check of course is not performed on the upper edge. Because some shapes with very evident lateral edges can satisfy the previous condition even if the lower edge is barely visible, an additional test is done on it, requiring that at least 30% of its pixels are black.

3.2 Tracking Phase

This phase is reached by the algorithm when a vehicle is found in the recognition phase. At this point, the information on presence and position of a vehicle is known. So, the first action that is performed is to check whether the vehicle recognized in the past is still in the scene. For this reason, the portion of grey-level image containing the vehicle in the past frame is compared with the content of the same rectangle in the last acquired image. Comparison, again, means correlation between the two portions of image inside the rectangles. Since a vehicle should be tracked also if it is slowly moving in the image, correlation with translated copies of the rectangle of interest found in the past are also computed. The box is translated in all directions, saving the corresponding correlation values. When all possible translations are considered, the best matching position suggests the motion direction of the vehicle, and the rectangle containing the target is accordingly moved. It is even possible that the rectangle needs to be resized, if the vehicle slowly gets closer or further, so, in the edges correlation sub-image, rows and columns near current edges are analyzed, and may become new edges if they have higher number of black pixels or a better placement. After translation and resizing, a new box is found. It should anyway be checked, so all tests described in the case of the recognition phase are performed.

To produce robust results, the presence of a vehicle is declared only after it has been recognized for a number of frames; in this way some spurious false positives are filtered out, with the small drawback that vehicles are found a little later than they could be. Once a vehicle has been found, it is tracked until it fails checks for five successive frames. This behavior is necessary to continue tracking the vehicle in some situations, when particular lighting conditions or shadows may produce some critical frames, causing at least one check to fail. For instance, when a vehicle goes under an overpass, the shadow causes the loss of the lower edge of the car. It was found that good results were found declaring the presence of a vehicle after it was detected for two frames.

Automatic Thresholds. Three thresholds used by the algorithm are particularly crucial and must therefore be carefully chosen; one is used to binarize the Sobel edge image, another to define how many times an edge should be present in the past to be in the edges correlation sub-image, and the last one is used in the correlation check. After some testing, values giving good results were found, but



(a) (b)

Fig. 3. In (a) a frame is processed with normally used thresholds. In (b) the same frame is processed using modified thresholds, which makes recognition possible.

there are still different values that could perform better in particular situations. So, an automatic adjustment of such thresholds has been implemented.

At each new frame, results of low-level processing are analyzed to understand if they bear poor information; in this case, the mentioned thresholds are modified trying to get better low-level images. If, for example, the edges correlation sub-image is composed by too few black pixels, like in Fig. 3 (a), the binarization threshold is lowered, as well as the number of frames in the past each edge must appear in to be visible in the edges correlation sub-image. Fig. 3 (b) represents the same frame as in (a) processed using these modifications. The threshold of the correlation check is modified when the target vehicle is particularly small. Such check, actually, often fails because each movement of the camera, even of a few pixels, becomes significant if related to the small number of pixels inside the rectangle of interest. Threshold corrections can improve edge detection, but also increase noise in the low-level images.

4 Results

The algorithm described so far was tested on some sequences acquired in highway environments, in different lighting conditions, for a total of more than 5 000 frames. When a preceding vehicle (car or truck) is present in the scene, the recognition rate is 70.53%. Otherwise, when there is no vehicle in the acquired scene, the false detection rate is only 5.71%. While the latter is completely satisfactory, some remarks can be made about the former, which may be improved. Some problems were encountered due to bad lighting conditions, such as at sunset, when images appear dark, like it can be seen in Fig. 3. In some cases the automatic thresholds are able to fix the problem, but in other cases this does not happen, and, in general, the recognition becomes harder.

4.1 Further Improvements

An open issue is the fact that the edges correlation sub-image still does not filter lane markers. Along bends, these become curve and may cause some columns to have a high number of nearby black pixels, leading to an erroneous evaluation of the rectangle of interest. Moreover, by lowering the number of frames used in the

corresponding computation the algorithm may be able to work also in contexts different from highways. An automatic recognition of the kind of environment may therefore be very useful to obtain the best behavior in different roads.

4.2 Processing Time

The computational time is a strong aspect of the algorithm. On a Pentium 4 processor, working at 2.80 GHz, with 1 GB RAM, the average and maximum processing times for each frame are 14 ms and 20 ms respectively, which allow the algorithm to work at a frequency of up to 50 Hz.

When the algorithm is in the recognition phase, the main part of time is used to compute correlation images (88%), while the remaining part is spent in processing edges (10%) and doing high-level tasks (2% only). In the tracking phase these proportions slighly change, but the computation of the correlation sub-image still takes the longest computational time (79%), while edges processing takes 10% of the time, while high-level now is 11% of total time. In the last case, high-level tasks are more time consuming than in the former because they need correlation too, when the rectangle of interest of the previous frame must be superimposed and best matched with the corresponding part of the last frame.

References

1. Abou-Jaoude, R.: ACC Radar Sensor Technology, Test Requirements, and Test Solutions. IEEE Trans. on Intelligent Trasportation System 4 (2003) 115–122
2. Denasi, S., Quaglia, G.: Obstacle Detection Using a Deformable Model of Vehicles. In: Procs. IEEE Intelligent Vehicles Symposium 2001, Tokyo, Japan (2001) 145–150
3. Fleischer, K., Nagel, H.H., Rath, T.M.: 3D-Model-based-Vision for Innercity Driving Scenes. In: Procs. IEEE Intelligent Vehicles Symposium 2002, Paris, France (2002)
4. Kuehnle, A.: Symmetry-based vehicle location for AHS. In: Procs. SPIE - Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS. Volume 2902., Orlando, USA (1998) 19–27
5. Broggi, A., Cerri, P., Antonello, P.C.: Multi-Resolution Vehicle Detection using Artificial Vision. In: Procs. IEEE Intelligent Vehicles Symposium 2004, Parma, Italy (2004) 310–314
6. Zeng, Z., Ma, S.: An Efficient Vision System for Multiple Car Tracking. In: Procs. IEEE 16th Intl. Conf. on Pattern Recognition. Volume 2. (2002) 609–612
7. Cao, G., Jiang, J., Chen, J.: An Improved Object Tracking Algorithm based on Image Correlation. In: Procs. IEEE Intl. Symp. on Industrial Electronics, Rio de Janeiro, Brasil (2003) 598–601
8. Betke, M., Haritaoglu, E., Davis, L.: Multiple Vehicle Detection and Tracking in Hard Real-time. In: Procs. IEEE IV Symposium'96, Tokyo, Japan (1996) 351–356

Statistical Displacement Analysis for Handwriting Verification

Yoshiki Mizukami[1], Katsumi Tadamura[1],
Mitsu Yoshimura[2], and Isao Yoshimura[3]

[1] Yamaguchi University
[2] Ritsumeikan University
[3] Tokyo Science University

Abstract. In this paper, it is assumed that each writer has his or her own statistics of handwriting displacement, therefore a statistical displacement analysis for handwriting verification is proposed. Here, a regularization method with the coarse-to-fine strategy computes the displacement function in questionable handwritten letters, and then it is normalized to remove the noisy displacement that arises from the position drift and scaling variation. Finally, the normalized displacement function and the statistics of displacement obtained in advance from registered authentic letters are used to calculate the distance from a standard handwritten letter to a questionable one. A fundamental simulation was conducted in order to evaluate the performance of the proposed method.

1 Introduction

Handwriting is one kind of biometrics and has played a major role in proving documentary evidence. As shown in the review of Plamondon et al. [1], many researchers are involved with this topic. Until now, several groups have tried to measure distance or similarities between a registered signature and a questionable signature after computing the displacement contained in the questionable one. For instance, de Bruyne and Forre[2] devised a linear deformation method for signature verification and Naske[3] suggested a block matching method. Mizukami et al.[4] computed the non-linear displacement in questionable signatures based on March's regularization algorithm[5, 6], which has been successfully applied to handwritten character recognition[7].

After reviewing previous attempts, it was noted that most of the researchers and engineers have been computing a displacement in order to eliminate it, while human handwriting experts are focusing on analyzing the tendency of the displacement. In this study, therefore, we propose a statistical displacement analysis method for handwriting verification based on an assumption that each writer has his or her own statistics of handwriting displacement, such as its average and variance of displacement. The following sections describe the details of the proposed method, the simulation results and the conclusion.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1174–1181, 2005.

2 Proposed Method

In the field of computer vision, March[6] suggested a regularization method for acquiring disparity in two stereo images. This study utilizes March's method and the coarse-to-fine strategy together to compute 2-dimentional displacements in questionable handwriting[4], normalizes the obtained displacement, and computes the distance based on the statistics of displacement to verify the questionable handwriting (See Fig. 1). In this section, first the computation of displacement with the coarse-to-fine strategy is described, and then the procedures for normalizing the computed displacement and measuring the distance are explained.

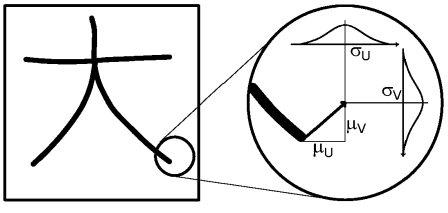


Fig. 1. statistics of displacement in handwriting

2.1 Displacement Computation Based on Coarse-to-Fine Strategy

The computation procedure of optimum displacement field between two isolated handwritten letters is described here. A handwritten letter is denoted as a binary image $f(x, y)$, where $1 \leq x \leq n_x$ and $1 \leq y \leq n_y$, another handwritten letter as $g(x, y)$. In the framework of regularization theory[5], the optimum displacement is given as a 2-dimensional functions $(u(x, y), v(x, y))$, which minimize the following functional $E(u, v)$,

$$E(u, v) = P(u, v) + \lambda S(u, v), \tag{1}$$

$$P(u, v) = \int\int \{f(x+u(x, y), y+v(x, y)) - g(x, y)\}^2 dxdy, \tag{2}$$

$$S(u, v) = \int\int \{(\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 + (\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial y})^2\} dxdy, \tag{3}$$

where the functional $P(u, v)$ represents the difference between f and g with taking into account the displacement function (u, v), the functional $S(u, v)$ represents the penalty of departure from smoothness in (u, v) and λ is a regularization parameter balancing them[8]. According to calculus of variation, the following Gauss-Seidel iterative equations give the optimum displacement function (u, v),

$$u^{[t+1]}(x, y) = \bar{u}^{[t]}(x, y) - \frac{1}{4\lambda} \frac{\partial f(x + \overline{u}^{[t]}(x, y), y + \overline{v}^{[t]}(x, y))}{\partial x} \times \{f(x+\bar{u}^{[t]}(x, y), y+\bar{v}^{[t]}(x, y)) - g(x, y)\}, \tag{4}$$

$$v^{[t+1]}(x,y) = \bar{v}^{[t]}(x,y) - \frac{1}{4\lambda}\frac{\partial f(x+\overline{u}^{[t]}(x,y), y+\overline{v}^{[t]}(x,y))}{\partial y} \times \{f(x+\bar{u}^{[t]}(x,y), y+\bar{v}^{[t]}(x,y)) - g(x,y)\}, \quad (5)$$

where the subscript t stands for the number of iteration and $(\bar{u}, \bar{v})$ is an average of four neighboring (u, v)'s.

We describe the coarse-to-fine strategy, which is adopted to keep its solution from being trapped by local minima and compute the displacement function (u, v) efficiently. The number of layers for the coarse-to-fine strategy is assumed to be K and two images in the k-th layer is referred as f_k and g_k, where $1 \leq k \leq K$. Note that f_K and g_K represent the original images of two handwritten letters. By applying the smoothing filter with size of 3×3 pixels to both of f_k and g_k and sampling at regular intervals, the coarser images, f_{k-1} and g_{k-1}, are generated. The same procedure is repeated until f_1 and g_1 in the first layer are obtained. The computation of the displacement function begins at the first layer. The obtained results are used as the initial values in the next finer layer. Consequently, the displacement function (u, v) with the same size of the original images is acquired.

2.2 Normalization Procedure for Computed Displacement

As described in Section 1, the uniqueness of the proposed method is to compute the displacement function and analyze it for handwriting verification. Since the computed displacement does not only contain the individual tendency in handwriting but also the effects of the position drift and scaling variation, we should remove such kinds of noisy effects from the computed displacement. Most of previous studies normalized the pattern on the image so as to fit the specified outer frame, while this study proposed a displacement normalization procedure that estimates the position drift and scaling variation according to the computed displacement. The details of the proposed normalization are described below.

It is assumed that (u', v') is the displacement caused by handwriting f's shift drift for g and that (z_x, z_y) is the horizontal and vertical scaling ratio of f for g, therefore the following equations are given,

$$u' = \frac{\sum g(x,y)u(x,y)}{n_p}, \quad (6)$$

$$v' = \frac{\sum g(x,y)v(x,y)}{n_p}, \quad (7)$$

$$z_x = \frac{n_p \sum g(x,y)u(x,y)x - \sum g(x,y)x \sum g(x,y)u(x,y)}{n_p \sum (g(x,y)x)^2 - (\sum g(x,y)x)^2} \quad (8)$$

$$z_y = \frac{n_p \sum g(x,y)v(x,y)y - \sum g(x,y)y \sum g(x,y)v(x,y)}{n_p \sum (g(x,y)y)^2 - (\sum g(x,y)y)^2} \quad (9)$$

where n_p is the number of pixels composing the handwriting g and given as $\sum g(x,y)$. We adopted a very simple idea, that is, if f has the horizontally

z_x-times scaled shape compared with g, the computed horizontal displacement $u(x, y)$ should increase with the gradient of z_x to x. The relationship of z_y and $v(x, y)$ is also the same as described here.

According to the obtained (u', v') and (z_x, z_y), the normalized displacement function (U, V) is given as

$$U(x, y) = u(x, y) - u' - z(x - \frac{n_x + 1}{2}), \tag{10}$$

$$V(x, y) = v(x, y) - v' - z(y - \frac{n_y + 1}{2}), \tag{11}$$

where z is the average of z_x and z_y. It means that the 2-dimensional scaling ratio of f for g is determined by balancing the horizontal and vertical scaling ratio.

2.3 Statistical Displacement Analysis for Handwriting Verification

The method of obtaining the statistics of displacement and applying them for calculating the distance of the questionable handwriting (See Fig.1 again) are described. It is assumed that N registered authentic handwritings are available in advance and that the n'-th handwriting, $g_{n'}$, is used as a standard one, where $1 \leq n' \leq N$. The average displacement function (μ_U, μ_V) and the variance (σ_U^2, σ_V^2) are given with the following equations,

$$\mu_U(x, y) = \frac{1}{N} \sum_{n=1}^{N} U_{n,n'}(x, y), \tag{12}$$

$$\mu_V(x, y) = \frac{1}{N} \sum_{n=1}^{N} V_{n,n'}(x, y), \tag{13}$$

$$\sigma_U^2(x, y) = \frac{1}{N-1} \sum_{n=1}^{N} (U_{n,n'}(x, y) - \mu_U(x, y))^2, \tag{14}$$

$$\sigma_V^2(x, y) = \frac{1}{N-1} \sum_{n=1}^{N} (V_{n,n'}(x, y) - \mu_V(x, y))^2, \tag{15}$$

where $(U_{n,n'}, V_{n,n'})$ is the normalized displacement function obtained in the case that the n-th and n'-th registered handwritings are used as f and g, respectively.

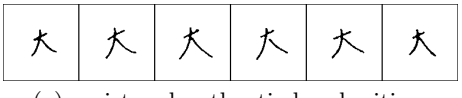
In order to verify a questionable handwriting f, after obtaining the normalized displacement function between f and g, the statistical distance d_{DA} is calculated with the following equation,

$$d_{DA} = \sum_{x,y} \frac{g(x, y)}{(1 - \alpha_{DA}) + \alpha_{DA}\sigma_U^2(x, y)} (U(x, y) - \mu_U(x, y))^2 + \sum_{x,y} \frac{g(x, y)}{(1 - \alpha_{DA}) + \alpha_{DA}\sigma_V^2(x, y)} (V(x, y) - \mu_V(x, y))^2, \tag{16}$$

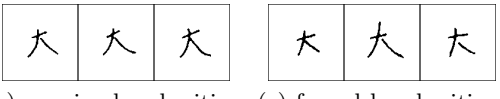
where α_{DA} is a parameter controlling the effect of (σ_U^2, σ_V^2). By comparing the obtained distance d_{DA} with the advance-specified threshold d_{th}, the questionable handwriting is judged if it is genuine or not.

3 Simulation Results

In order to evaluate the performance of the proposed method, a fundamental simulation was conducted. One volunteer offered twenty authentic handwritten letters for registration, $G_{reg} = \{g_{reg,1}, \cdots, g_{reg,20}\}$, and ten genuine handwritten letters for the evaluation, $F_{gen} = \{f_{gen,1}, \cdots, f_{gen,10}\}$, while ten other volunteers also offered ten skillfully forged handwritten letters for the evaluation, $F_{fog} = \{f_{fog,1}, \cdots, f_{fog,10}\}$. They were allowed to practice imitating one of the genuine handwritings and write a forged handwriting while looking at the genuine letter. Handwritings used in our experiment are a subset of the handwritten signature database produced in M. Yoshimura and I. Yoshimura's research[9]. The size of the image was 128×128 pixels. The horizontal and vertical widths of all the handwritten letters on the image were shorter than 100 pixels. Each letter was shifted on the image so that the center of gravity could be located at the center of the image. Figure 2 illustrates the examples of a handwritten letter. It is a typical Chinese character that means 'big' or 'large' in English.



(a) registered authentic handwriting.

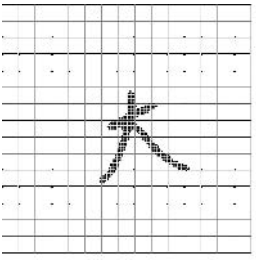


(b) genuine handwriting. (c) forged handwriting.

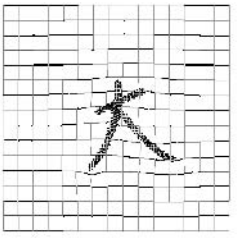
Fig. 2. examples of Chinese handwriting sets

First we acquired the statistics described in 2.3, that is, (μ_U, μ_V) and (σ_U^2, σ_V^2). The first registered authentic handwriting, $g_{reg,1}$, was used as a standard one. Figure 3(a) and (b) illustrate the standard one and the handwriting deformed with the average normalized displacement function (μ_U, μ_V). On the other hand, Fig. 3(c) and (d) illustrate the variance (σ_U^2, σ_V^2) of the normalized horizontal and vertical displacement functions, respectively, where more dark area means that its variance is larger. Note that, as shown in Eq. 16, only the statistics whose pixel is on the handwritten letter is utilized for calculating the distance d_{DA}.

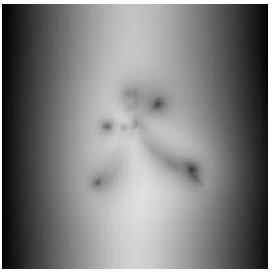
Next the procedure for calculating the statistical distance d_{DA} of a questionable handwriting is described. The regularization method with the coarse-to-fine



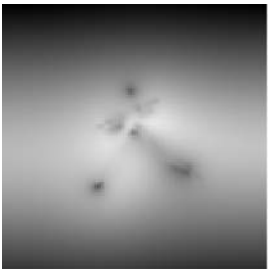
(a) standard handwriting, $g_{reg,1}$.



(b) average displacement, (μ_U, μ_V).



(c) variance of horizontal displacement, σ_U^2.



(d) variance of vertical displacement, σ_V^2.

Fig. 3. standard handwriting and displacement statistics

strategy computed the displacement function in the questionable handwriting, and then it was normalized so as to remove the noisy displacement that arose from the position drift and scaling variation. Finally, according to the displacement statistic obtained in advance from registered authentic signatures, the statistical distance d_{DA} between the questionable and standard handwritings is calculated. In this simulation, K was set to 5 and λ's were set to 5.0×10^{-3}.

Most of the previous studies evaluated the performance of the signature verification methods from the viewpoint of two types I and II error ratio, i.e., error ratio for false judgment on genuine signatures and that for false judgment on forged signatures. In this study, however, we employed the degree of separation to keep the performance evaluation from being complicated due to the configuration problem of the threshold d_{th}. The degree of separation (DoS) is defined by the following equation,

$$\eta = \frac{(\mu_{gen} - \mu_{for})^2}{\sigma_{gen}^2 + \sigma_{for}^2}, \tag{17}$$

where μ_{gen} and μ_{for} are the average of the measured distances from the standard handwriting to the genuine and forged ones, respectively. In addition, σ_{gen}^2 and σ_{for}^2 are the variance to each of them, respectively.

For the comparison, the result of a conventional template matching is also shown, where the average image $\mu_g(x, y)$ and the variance $\sigma_g^2(x, y)$ was computed based on twenty registered authentic handwriting g_{reg}'s. This conventional method defines the distance d_{TM} by the following equation,

$$d_{TM} = \sum_{x,y} \frac{1}{(1-\alpha_{TM}) + \alpha_{TM}\sigma_g^2(x,y)} (f(x,y) - \mu_g(x,y))^2, \tag{18}$$

where α_{TM} is a parameter controlling the effect of $\sigma_g^2(x,y)$.

Figure 4 shows the relationships of α and DoS of the proposed and conventional method. In the case of $\alpha = 0$, the DoSs were 2.27 and 1.61, respectively. The proposed method gave the best DoS of 2.44 at $\alpha_{DA} = 0.75$, while the conventional method gave that of 1.72 at $\alpha_{TM} = 0.97$. These results implied the superiority of the proposed method.

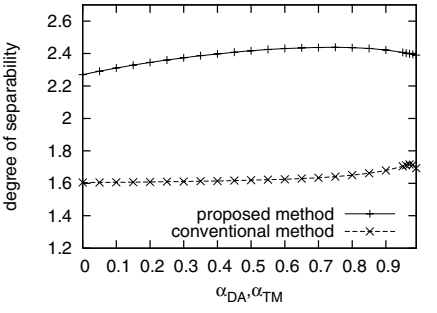


Fig. 4. degree of separation with proposed and conventional method

In order to investigate the improvement of DoS by combining the proposed method with the conventional one, the combined distance described below was introduced.

$$d_{mix} = (1-\beta)d_{TM} + \beta d_{DA}, \tag{19}$$

where β is a parameter balancing the proposed and conventional distance.

Figure 5 illustrates the relationship between β and DoS. The best DoS was 2.56 at $\beta = 0.9$, which means that this combination can give further performance.

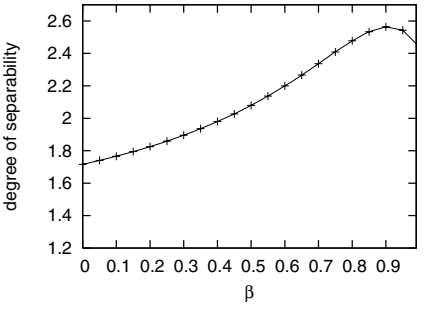


Fig. 5. degree of separation with using proposed and conventional method together

4 Conclusion

In this paper, it was assumed that each writer has his or her own statistics of handwriting displacement, therefore a statistical displacement analysis for handwriting verification was proposed. Here, a regularization method with the coarse-to-fine strategy computes the displacement function in questionable handwritten letters, and then it is normalized to remove the noisy displacement that arises from the position drift and scaling variation. Finally, the normalized displacement function and the statistics of displacement obtained in advance from registered authentic letters are used to calculate the distance from a standard handwritten letter to a questionable one.

In order to evaluate the performance of the proposed method, a fundamental simulation was conducted. One volunteer offered twenty authentic handwritten letters for registration and ten genuine handwritten letters for the evaluation, while ten volunteers also offered ten skillfully forged handwritten letters for the evaluation. An indicator, degree of separation, was employed for studying how far the method separates the genuine and forged handwritings. The simulation results show that considering the statistics in handwriting displacement is effective for handwriting verification, and that the use of the proposed method with the conventional template matching method together gives further performance.

References

[1] Plamondon, R. and Lorette, G. 1989. Automatic signature verification and writer identification – the state of the art. *Pattern Recognition*, 22(2):107–131.

[2] de Bruyne, P. and Forré, R. 1986. Signature verification with elastic image matching. *Proc. International Carnahan Conference on Security and Technology*, pages 113–118.

[3] Naske, R.-D. 1982. Writer recognition by prototype related deformation of handprinted characters. *Proc. 6-th International Conference on Pattern Recognition*, 2:819–822.

[4] Mizukami, Y., Yoshimura, M., Miike, H., and Yoshimura, I. 1999. An off-line signature verification system using an extracted displacement function. *Proc. 5th ICDAR*, 1:757–760.

[5] Poggio, T., Torre, V., and Koch, C. 1985. Computational vision and regularization theory. *NATURE*, 317(6035):314–319.

[6] March, R. 1988. Computation of stereo disparity using regularization. *Pat. Recog. Let.*, 8(3):181–188.

[7] Mizukami, Y. 1998. A handwritten Chinese character recognition system using hierarchical displacement extraction based on directional features. *Pattern Recognition Letters*, 19(7):595–604.

[8] Horn, B. and Schunck, B. 1981. Determining optical flow. *Artificial Intelligence*, 17:185–203.

[9] Yoshimura, I. and Yoshimura, M. 1994. Off-line verification of Japanese signature after elimination of background patterns. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(3):693–708.

3D Functional Models of Monkey Brain Through Elastic Registration of Histological Sections

Fabio Bettio[1], Francesca Frexia[1], Andrea Giachetti[1,2], Enrico Gobbetti[1], Gianni Pintore[1], and Gianluigi Zanetti[1]

[1] CRS4 - c/o POLARIS, Edificio 1, Loc. Piscinamanna 09010 Pula (CA), Italy
giach@crs4.it
[2] Universitá degli Studi di Cagliari, Dipartimento di Matematica e Informatica, via Ospedale, 72, 09124 Cagliari

Abstract. In this paper we describe a method for the reconstruction and visualization of functional models of monkey brains. Models are built through the registration of high resolution images obtained from the scanning of histological sections with reference photos taken during the brain slicing. From the histological sections it is also possible to acquire specifically activated neurons' coordinates introducing functional information in the model. Due to the specific nature of the images (texture information is useless and the sections could be deformed when they were cut and placed on glass) we solved the registration problem by extracting corresponding cerebral cortex borders (extracted with a snake algorithm) and computing an image transform from the deformation linking them. The mapping is modeled as an affine deformation plus a non-linear field evaluated as an elastically constrained deformation minimizing contour distances. Registered images and contours are used then to build 3D models of specific brains by a software tool allowing the interactive visualization of cortical volumes together with the spatially referenced neurons classified and differently colored according to their functionalities.

1 Introduction

The study of anatomy and functionality of the cerebral cortex is one of the most important research fields in neuroscience. Non-invasive methods to study human brains still have strong limitations in spatial and temporal resolution and in the possibility of correlating anatomical and functional data. Neuroscientists are therefore forced to analyze histological sections of animal brains to understand more precisely morphology and functionality. A typical method to acquire data is the slicing of the frozen brain mass creating histological sections placed on glass plates. From these sections it is possible to recover the position of neurons involved in particular activities that can be visualized on the slices through the use of different tracers. To recover 3D models and locate neurons in a 3D spatial reference it is, however, necessary a complex processing of the acquired data involving non rigid registration of images. To visualize in an effective way the

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1182–1189, 2005.

data is then necessary to use advanced visualization tools reconstructing surfaces from registered image stacks and rendering them with neuron information added.

In this paper we describe specific registration algorithms and tools for model visualization realized in the framework of the URBAN project, funded by the Italian Ministry of University and Research aimed at the reconstruction of this kind of functional models. Through the use of an ad hoc image registration method and modern visualization techniques, the project allowed neuroscientists to have a clear view of spatially referenced activated neurons over the specific morphology of the brain cortex. Adding cortex segmentation, they can also visualize selectively different functional regions of the brain and different kinds of activated neurons.The paper is organized as follows: Section 1 describes the image registration problem, Section 2 presents the approach chosen, Section 3 the registration results, Section 4 model reconstruction and visualization, Section 5 a short discussion.

2 Reconstruction Framework

Image processing techniques can then be used to recover functional models from these glass plates. Digital images can, in fact, be recovered from them with high resolution planar scanners and, being known the slice spacing, voxelized or surface models of the brain with referenced neurons can be built from image stacks. The problem is that the locations of corresponding structures in consecutive images of a stack are not correctly matching because the manual procedure of slice fixation makes slices shifted and deformed by local stretchiness. In order to realign and deform correctly the images and recover therefore the correct spatial location of all the brain structures, digital photos in a fixed reference system are taken. For each glass plate a corresponding photo is thus available and registering the glass plates images with it allows the reconstruction of morphological and functional models of the brain. The reconstruction problem is therefore mainly a non rigid image registration problem, where the main difficulty is due to the lack of texture correspondence between images: only feature points or contours can be used.

3 Slices Registration

3.1 Image Registration: An Overview

Image registration is a typical task of computer vision. It consists in finding a correspondence map between a pair of images, similarly to what happens in optical flow or stereo matching problems. Here, however, the two images usually represent a 2D textured planar surface, not a collection of 3D objects in space, and the correspondence field to be found represents a deformation of the object seen – due to the use of different imaging modalities or, e.g., inter–subject difference in the anatomy of the tissue imaged – rather than the composition of camera motion and object displacement.

The deformation field linking images acquired using different imaging modalities are, usually, global and with a simple structure, i.e., can be described as affine or polynomial transforms that depend only on changes in sensors orientation and positions and that can be easily identified by imposing the correspondence between few points in the images. Several techniques have been proposed to find points matching, ranging from invasive methods based on the insertion of fiducial markers during the acquisition process, to computer vision algorithms such as landmark recognition, correlation. A complete classification of registration methods used in medical image processing can be found in [7]. In [11] a comparison of results is also presented and a more recent survey can be found in [12]. Things are much more complex when one wants to find how to link images of things that are actually different even if related, like corresponding structures of different subjects, stretched objects, etc. This is the case for example of inter-subject registration of CT or MRI sections acquired from different patients or intra subject scans at different times and it is also the case of our interest. A typical approach for this task is to apply an iterative elastic deformation [3,10], driven by the correspondence of control points "recognized" in the two images. Other methods found in literature are piecewise linear transforms [9], or physically based deformations, recently improved by Hagemann et al. [5]. Impressive results have been obtained by Chu and Rangarajan [4] using an iterative method based on a Thin Plate Spline deformation model. The method is fast and robust because of soft-assignment of points correspondence and the possibility of automatically detecting and discarding outliers. Contour based image registration is, however, a slightly different problem, due to the fact that there is no gray level correspondence between the two images and the input of the algorithm are couple of corresponding chains or parametric curves in the two images. The few existing approaches specific to this problem are based on contour points sampling and matching and subsequent point based registration.

A simple algorithm was proposed in [6]. It consists in resampling the images at the same resolution and extract contours as chain codes. Point matching is performed as contours pixel matching from minimization of chain codes' difference. A more advanced approach that is closer to ours was proposed by Davatzikos et al.[2]. They first extract contours from the two images with snakes, then find corresponding points and use an iterative Gauss-Seidel method to solve a dynamical system where the image pixels are shifted driven by the contour points correspondence and by elastic forces (using a spring-mass model) avoiding pixel position folding. The method has problems in handling large deformations and can have problems due to local contour deformations.

3.2 The Method Implemented: Contour Driven Iterative Deformation

The method we designed uses of a three step transform handling large deformations and accounting for local contour stretching. Before describing algorithms, let us give a formal description of the problem. We have a target image T, and an input image I. Finding the correspondence of one or more contours extracted in

T, c_T with contours in I, c_I, we want to recover the non-linear transform giving for each pixel $p = (i, j)$ in I, the corresponding pixel $P = (U(i, j), V(i, j))$ in the reference system of T, creating a new image R that we call the "registered" image. Corresponding contours are closed parametric curve $c_T(k, s)$, $c_I(k, s')$ where k is the number of the corresponding contour and s, s' the contour parameter, that in the discrete approximation correspond to the contour point label. The correspondence of contours' points is supposed to drive the image deformation that maps $c_I(k, s)$ into $c_T(k, s)$ for each k,s.

Contour Extraction. Contours are extracted from raster images with a snake algorithm [8]. In our implementation contours are simple chains of points. Forces driving their iterative evolution are: standard elastic/rigid forces (computed with finite differences), an inflating or deflating force making the curve move toward the region limits, image force based on an RGB threshold (contour evolution is stopped if the distance of the color value from the central on in the RGB space is lower than a fixed threshold), a color edge force. Automatic resampling keeps the point distribution constant (the number of points or the point spacing are kept constant). In our experiments, functional data were often acquired with another method, i.e. using a particular device allowing neuroscientists to inspect through a microscope glass plates and to manually trace borders and mark neuron positions. Contours' and neurons' coordinates were in these cases given us as a vector image describing the contours as polylines and neuron positions as 2D coordinates. In this case the functional model can be recovered in the same way, registering the functional data with respect to the same reference photographic images. The registration procedure for this data is the same,only the contour to be mapped on the reference image is directly selected on the file instead of computed with the snake and the transform obtained is applied then to the neurons instead of to the pixel array.

Point Matching. When couples of corresponding contours are found, we resample them with the same number of point, equally spaced $P_1(i), P_2(i)$. Then we match labeled points with a nonlinear transform considering the possibility of local stretching of the contours. The procedure adopted is the following: first a global correspondence of the original equally spaced nodes of the original contour parameterization is done by finding the integer label shift of the point of P_2, s minimizing the function: $f = \sum_{i=1}^{N}(w(P_1(i)) - w(P_2((i + s) mod N)))^2$, where w is a function describing locally the curve not depending on the reference system. We took our w as a linear combination of the squared distance of the point from the center of mass of the curve divided by the surface enclosed by the curve and the local curvature. Then feature points are detected in the contours extracted from the target and the input images. If two feature points of the same kind are found among the points of the input and target contours within a small label range, a correspondence between the two points is set. When all the corresponding features are found, the input contour is resampled giving the same label to the corresponding feature point and resampling the two contours between consecutive features points with the same number of equally labeled points. Features considered are:

-Local maxima of curvature larger than a threshold.
-Local minima and maxima of the distance from the center of mass.

Linear Transform. The first step in the registration procedure consists of a linear transformation. Writing its equation for points of corresponding contours with the same label an overconstrained system is obtained, and its least-squares solution gives the best linear transformation mapping the input image pixels into the target image pixels. The linear transform parameters are saved as well as the displacement field mapping the input image into the linearly registered one. The field and the new image are built by creating a pixel grid of target image size and computing for each pixel location the vector mapping the target pixel into the input image. A bilinear interpolation is finally used to compute the color components of the pixel from the four input image pixels closest to these non-integer transformed coordinates.

Fast Iterative Elastic Transform. A non linear correction is used as the second registration step. It is an iterative elastic method with some features in common with the method used in [2] The basic idea is that pixels are supposed to be joined by springs to their 4 neighbors keeping them close, while pixels corresponding to matching contours are shifted according to the field defining the contours correspondence. The input image is the linearly registered one and the target image is the same as before. We create then a registered image of the target image size. The goal is to put in each pixel of this image values of the linearly registered one at a transformed location. The pixel transformations are computed through an iterative algorithm (similar to the one described in [2]) minimizing with a greedy algorithm the elastic potential with the constraints of zero motion at image borders and reassigning periodically input shifts at contour pixels. To compute these shifts we apply a Bresenham rasterization of the target contour and we assign to each pixel obtained a shift vector equal to its distance from the linearly registered corresponding contour. To speed up the computation, shifts of pixels inside the contour are not initialized as null vectors, but with x(y) values equal to a weighted average of the two closest border shifts along the x (y) direction. Weights are inversely proportional to the corresponding border points distance.

This procedure initializes the system closer to the equilibrium position. Finally pixel shifts is iteratively calculated from the spring-mass system reassigning periodically the distance values to the border pixels. Iterations are stopped when non border pixel shifts are lower than a fixed threshold.

4 Experimental Results

Tests on Registration. We first applied registration algorithms on two test image pairs, the first simulating a linear deformation and the second adding to a known linear transform some local non-linear effects similar to those coming from the acquisition of a histological sample (i.e with local stretching and irregularities). Recovered and ground-truth displacement fields have finally been compared

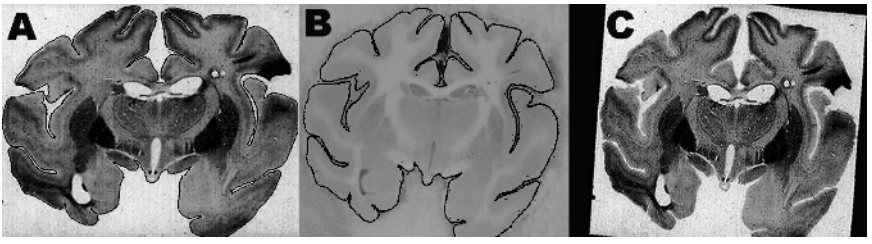


Fig. 1. Results on a real image pair. A,B: input and target images (and superimposed contour in black). C: registration results.

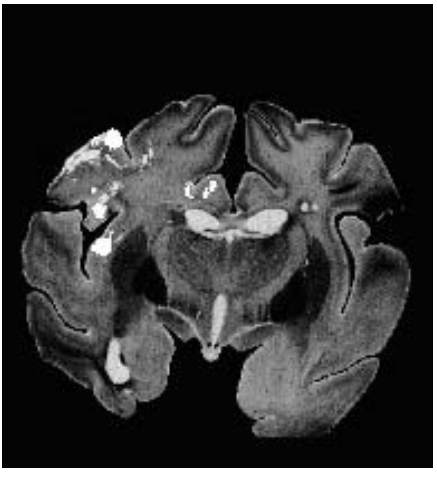


Fig. 2. Image representing more than 4000 classified neurons (bright spots) from a microscopy acquisition mapped over the registered slice texture

with standard optical flow difference measures [1] with the displacement fields computed by the registration algorithms. Experiments on the linearly deformed pattern have shown the advantages of using the feature based contour point sampling instead of the uniform contour sampling used in [2]. The average error in the deformation field for the internal pixels of the shape is decreased from an average unsigned difference of 1.3 to 0.3 and from an average Barron angular distance of 0.5 to 0.1, i.e. there is a great improvement in the registration accuracy due to the feature based resampling step. Tests performed also on the nonlinear synthetic deformation showed then that the minimization of point/contour distances provides better results than the minimization of point-point distances.

Registration of real images with our final method obtain qualitatively good results. Fig. 1A shows a reference photo acquired during the sectioning of a frozen monkey brain. Fig. 1B shows the corresponding scanning of the histological section. Contours semi-automatically recovered with deformable contours are represented superimposed to the corresponding image. It must be noted that

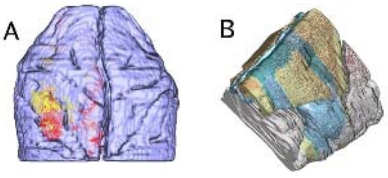


Fig. 3. Left: Brain reconstruction rendering with two neuron classes displayed in different colors. Right Brain reconstruction rendering with seven functional cortex section painted in different color and only one neuron class displayed.

the quality of the scans is not always satisfactory, so that automatically computed contours should be in some cases checked and locally deformed through a user friendly interface. Couple of corresponding images and contours is then registered with the method described in order to build the morphological brain model. Fig.1C shows the results of the registration algorithm based on linear deformation and fast elastic adaptation. Every histological section has been acquired at a resolution of 600 dpi which corresponds to an actual pixel size of about 70x70 microns. Neurons were acquired with the microscopy tool at University of Parma. Fig.2 shows an example of registration of classified neurons mapped over the corresponding slice textures. Both neurons and texture are mapped with the presented technique over the reference photos coordinates.

Model Reconstruction and Visualization. From the realigned images it is possible to build precise voxelized or surface models of the brain (linking borders in consecutive slices). We implemented an user friendly interface allowing the visualization of 3D reconstructed brain models. The main operations performed by the system are model reconstruction, using different algorithms such as isosurface extraction or point splatting technique, and multipass rendering, to display interactive selected ranges of labeled cortical neurons. Figure 3A shows a reconstructed brain cortex surface, rendered in an user friendly interface proving the neuroscientists useful tools for the functional analysis. Spatially referenced neurons are represented as dots painted through a colormap representing its functionality (here we see two types in red and yellow). The interactive tool allows the user to visualize only selected neurons. Neurons are modeled as emissive particles, while the brain material is considered as a semi-transparent medium with constant absorption coefficient. Absorption is computed based on distance from one neuron to the brain surface closest to the eye. In the case of Fig.3B, different functional regions of the brain have been segmented and represented in different colors, with superimposed neurons of a single type represented in red.

5 Conclusions

We presented a framework for the reconstruction of functional models of monkey brain. It is based on a contour based image registration method that is an improvement of other approaches found in literature and on an interactive tool

for model analysis and visualization. Results have been obtained within the URBAN project, in collaboration with the Dept. of Neuroscience of the University of Parma, aimed at the reconstruction of 3D functional maps of primate brains, funded by the Italian Ministry of Instruction, University and Research.

References

1. J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt. Performance of optical flow techniques. *CVPR*, 92:236–242.
2. C. Davatzikos, J. Prince, and R. Bryan. Image registration based on boundary mapping. *IEEE Transactions on Medical Imaging*, 15:112–115, 1996.
3. Matthieu Ferrant, Simon K. Warfield, Charles R. G. Guttmann, Robert V. Mulkern, Ferenc A. Jolesz, and Ron Kikinis. 3d image matching using a finite element based elastic deformation model. In *MICCAI*, pages 202–209, 1999.
4. A. Rangarajan H. Chui. A new point matching algorithm for nonrigid registration. *Comput. Vision Image Und.*, 89:109–111, 2003.
5. A. Hagemann, R. Stiel, and U. Spetzger. Non-rigid matching of tomographic images based on a biomechanical model of the human head. In *SPIE Medical Imaging '99*. 1999.
6. H. Li, B.S. Manjunath, and S.K. Mitra. A contour-based approach to multisensor image registration. *IEEE Trans on Image Processing*, 4(3):320–334, March 1995.
7. J. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.
8. D.Terzopoulos M.Kass, A.Witkin. Snakes: Active contour models. *Int. Journal of Computer Vision*, 1:321–331, 1987.
9. A. Pitiot, G. Malandain, E. Bardinet, and P. Thompson. Piecewise Affine Registration of Biological Images. In *Second International Workshop on Biomedical Image Registration WBIR'03*, 2003.
10. T. PM and T. AW. Elastic image registration and pathology detection, 2000.
11. J. West, J. Fitzpatrick, M. Wang, B. Dawant, C. Maurer, R. Kessler, and R. Maciunas. Comparison and evaluation of retrospective intermodality image registration techniques, 1996.
12. B. Zitova and J. Flusser. Image registration methods: a survey. *IVC*, 21(11):977–1000, October 2003.

An Application of Neural and Probabilistic Unsupervised Methods to Environmental Factor Analysis of Multi-spectral Images

Luca Pugliese[1], Silvia Scarpetta[3], Anna Esposito[1,2], and Maria Marinaro[1,3]

[1] IIASS, Istituto Internazionale per gli Alti Studi Scientifici "E.R.Caianiello", Via G.Pellegrino, 19 – Vietri sul Mare - Salerno
{iiass.luca@tiscali.it, iiass.annaesp@tin.it
[2] Dipartimento di Psicologia, Seconda Università di Napoli, Via Vivaldi 43, Caserta
[3] Dipartimento di Fisica "E.R.Caianiello", Università degli Studi di Salerno, Via S.Allende, Salerno, Italy and INFM and INFN Sezione di Salerno, Italy
silvia@sa.infn.it, marinaro@sa.infn.it

Abstract. In this paper we test the performance of two unsupervised clustering strategies for the analysis of LANDSAT multispectral images of the Temples of Paestum Area in Italy. The classification goal is to identify environmental factors (soils, vegetation types, water) on the images, exploiting the features of the seven LANDSAT spectral bands. The first strategy is a fast migrating means technique based on a Maximum Likelihood Principle (ISOCLUST algorithm), and the second is the Kohonen Self Organizing Map (SOM) neural network. The advantage of using the SOM algorithm is that both the information on classes and the similarity between the classes are obtained (since proximity corresponds to similarity among neurons). By exploiting the information on class similarity it was possible to automatically colour each cluster identified by the net (assigning a specific colour to each of them) thus facilitating a successive photo-interpretation.

1 Introduction

The analysis of remotely sensed multispectral data is of great interest for improving the knowledge of the Earth surface and remarkably contributes to the development of policies for planning and monitoring environmental resources [12]. The standard approach to the analysis of such images involves the grouping of image data into a finite number of discrete clusters or classes that identify the distribution, over the land, of environmental factors such as soils, vegetation, urban areas, and rivers. For several decades, such clustering has been implemented using classical statistical approaches, mostly based on the Maximum Likelihood Principle (MLP) [1-2], assuming that clusters can be modelled as a multivariate normal distribution. However, geographical phenomena are not randomly distributed in nature and are not always displayed in the image with a normal distribution. Therefore, other methods have been suggested to overcome their limitations, among which Artificial Neural Networks (ANN). Recent developments in the field have shown that supervised NN algorithms

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1190 – 1197, 2005.

are able to perform a better classification than statistical approaches due to the fact that no assumption is made on the cluster distribution in the image data [3-5]. However supervised classification with neural networks requires labelled data that generally are not available and therefore several authors have proposed methodologies based on unsupervised techniques and demonstrated their effectiveness in multispectral satellite images classification to land-cover categories [6], [7]. At the light of these last reported studies and with the aim to study the possibility to use remotely sensed data for the mapping of archaeological features we investigate the performance of two unsupervised techniques for land-cover classification in the Piana del Sele (Peastum), Italy. The present paper is organized as follow: Section 2 describes the area of interest and the image data available; Section 3 and Section 4 briefly report on the two selected unsupervised strategies and on the results obtained; and finally Section 5 reports our considerations and evaluation on the two proposed strategies.

2 Data Source and Study Area

A LANDSAT 7 Enhanced Thematic Mapper (ETM+) image from January 2003 was analyzed for this study using IDRISI Kilimanjaro geo-analytic and image processing system (http://www.clarklabs.org). The scene covers the south part of Campania region (Italy), specifically the area named *Piana del Sele* identified in the UTM (Universal Transverse Mercator) reference system, zone 33 North, with the following coordinates: Upper Left Corner(m): (498728.6411429; 4468395.1036465); Lower Right Corner(m): (507790.7439107; 4478689.1775624).

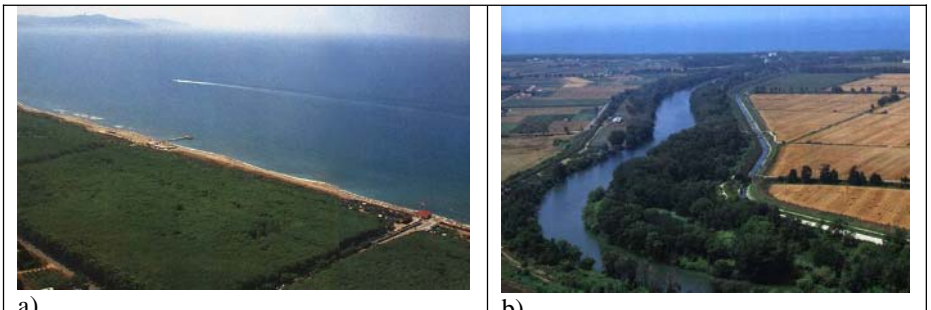


Fig. 1. a) Coastal zone of Paestum; b) Sele river (SA, Italy)

This area is primarily an agricultural land (no mountains are situated in the area) with major crops including corn, soybeans, grain, tobacco and canning vegetables, extensively irrigated by the water of the main river of the area, the River Sele. The coastal zone is mostly occupied by pine wood. On the area is also situated an ancient Greek archaeological site that preserves ruins dated 600 BC. During the last 60 years, an urbanization phenomenon has arisen in the area, producing a sparse dissemination of urban fabric and other human artefacts, especially for farming practice (greenhouses). Two aerial photos of zone are shown in Figure 1. The site contains many types of land

cover, among which water, urban areas, irrigated vegetation, unproductive terrain, agricultural areas with both permanent and seasonal crops, natural grassland, and conifers. The urban area is made up of discontinuous fabric mixed with vegetation. Besides, agricultural lands, natural grassland, and artificial surfaces are highly mixed, making difficult the land cover classification in some cases , especially in consideration of the ground resolution reported below.

LANDSAT 7 carries the Enhanced Thematic Mapper Plus instrument – a nadir-viewing multispectral scanning radiometer, providing image data of the Earth's surface via eight spectral bands (TM1, TM2, TM3, TM4, TM5, TM6, TM7, TM8). The bands are for the visible (TM1, TM2, TM3), the near infrared (TM4), the mid infrared (TM5, TM7), and the thermal infrared (TM6) regions of the electromagnetic spectrum, as well the panchromatic region (TM8). Table 1 lists the ETM+ bands, spectral ranges, and nominal ground resolution.

Table 1. LANDSAT 7 ETM+ bands, spectral ranges, and ground resolution (from Eurmoimage©)

Band Number	Spectral Range (μm)	Ground Resolution (m)
TM1 (Blue -Visible)	0.450 – 0.515	30
TM2 (Green - Visible)	0.525 – 0.605	30
TM3 (Red - Visible)	0.630 – 0.690	30
TM4 (Near Infrared)	0.760 – 0.900	30
TM5 (Mid Infrared)	1.550 – 1.750	30
TM6 (Thermal Infrared)	10.42 – 12.50	60
TM7 (Mid Infrared)	2.080 – 2.350	30
TM8 (Panchromatic)	0.520 – 0.900	15

Data are quantized at 8 bits. The size of an image, for each band, is 286 lines *x* 105820 pixels. We used seven of these bands as input to the two unsupervised techniques to be able to include all the spectral ranges in the analysis.

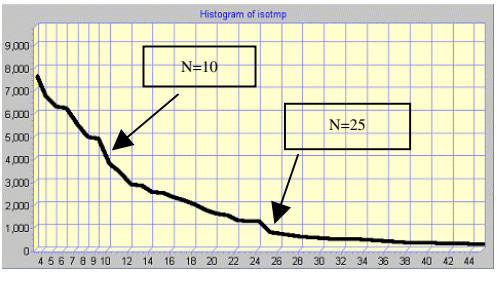


Fig. 2. One-dimensional histogram produced by the first step of the ISOCLUST module. The x-axis reports the number of the clusters identified by the histogram peak procedure and the y-axis reports the number of pixels in each cluster.

Therefore only the panchromatic band (TM8) was excluded, since it covers the same spectral range of TM2, TM3 and TM4. The seven images used as input data were acquired by EUROIMAGE© (http://www.eurimage.com/) and geometrically corrected at level 1G using a Nearest Neighbour re-sampling algorithm [8]. A COST [9] atmospheric correction was performed by the authors on the bands TM1 and TM2 due to the presence of haze. No processing was performed on the images.

3 ISOCLUST Algorithm Description and Results

ISOCLUST stands for Iterative Self Organizing CLUSTering and is a module proposed in the IDRISI Kilimanjaro image processing system implementing an algorithm that performs unsupervised classification. The algorithm is essentially based on an iterative optimization clustering procedure. Its originality lies in the definition of the initial condition (seeding step) from which the iterative process goes on to the final classification.

Normally, the seeding step is a random or systematic segmentation of the image to be classified, whilst the ISOCLUST module performs a true one-shot clustering process based on a histogram peak technique proposed by Richards [8]. For each input file, the procedure computes the histogram according to a specified number of gray levels. These histograms are then merged together creating a multidimensional histogram where pixels are accumulated. Peaks are then identified on this multidimensional histogram as those locations with a frequency higher than that of all the non-diagonal neighbors except one. Each peak with its neighborhood identifies a cluster and a one-dimensional histogram is plotted where the clusters identified by the histogram peak procedure, are ordered according to the number of pixels accumulated in each of them. Figure 2 shows the clusters identified in our experiment. The x-axis reports the number of the clusters identified by the histogram peak procedure and the y-axis reports the number of pixels in each cluster. In order to fix the number N of clusters needed for our classification significant breaks are identified in the curve in Figure 2, presumably representing major changes in the level of details for the description of the scene. For example, in our case, we identified in Figure 2 two major breaks at N=10 and at N=25 clusters. These values specify the number of clusters to be retained for the subsequent refining iterations of the ISOCLUST procedure. For each cluster identified in the first seeding step, a set of spectral signatures with their variance/covariance matrices are calculated and the pixels are regrouped in N new clusters modeled according to the Maximum Likelihood Principle. Due to the efficiency of the seeding step, very few iterations are required to produce a stable result. In our case only three iterations were performed. The ISOCLUST results for N=25 and N=10clusters are displayed in Figure 3.

The photo interpreter can easily recognize in Figure 3a) the coast and the sea is described by four different clusters (or colours) that are likely to represent differences in water turbidity, sea level and temperature (an information contained in the TM6 band, see Table 1). Starting from the coastline and proceeding inward, it is possible to identify the cluster describing the sea shore (red pink) that also contains the big regular structure visible in the upper right of Figure 3a, other similar smaller structures, and urban areas. The cluster groups together sea shore, and greenhouses and buildings

(artificial surfaces). Two clusters (indigo and bright blue) are descriptive of the pine wood and the River is identified in yellow (the upper left part of Figure 3a). However, all the other characteristics of the land are confused and difficult to be detected without the help of ancillary data.

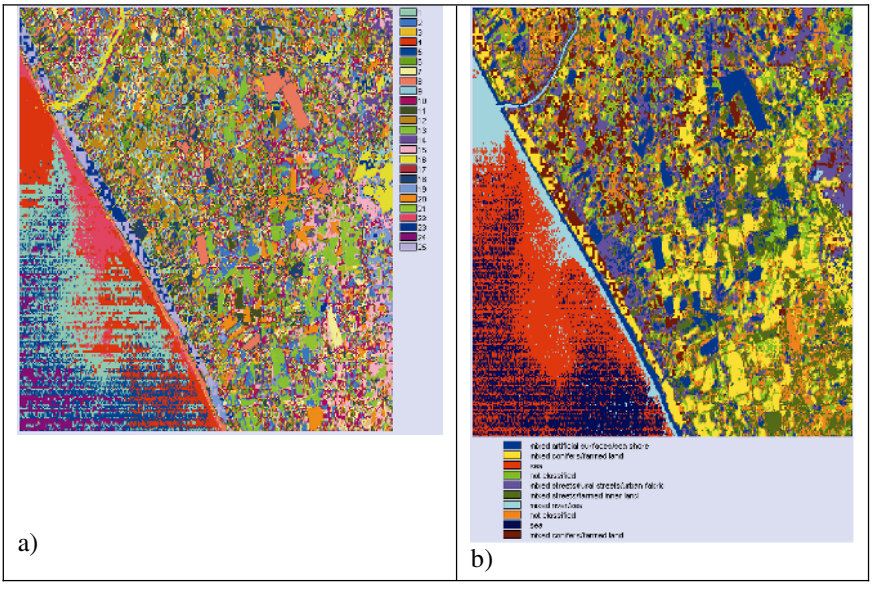


Fig. 3. Results obtained with Idrisi ISOCLUST algorithm using 25 spectral classes

To be able to coarsely detect the main characteristics of the land, a second experiment involving only N=10 clusters was performed. Results are displayed in Figure 3b showing that a reduced number of clusters worsens the quality of the clusterization. As it could be noticed in the legend, except for the sea, all the other classes are mixed or unrecognizable.

4 SOM Algorithm Description and Results

The Self-Organizing Map (SOM) [10] is an unsupervised neural network algorithm that has turned out to be an efficient tool, in various applications, for data exploration tasks. It carries out a nonlinear mapping of input data onto a two-dimensional grid. The mapping preserves the most important topological and metric relationships of the data. The SOM map consists of a regular grid of processing units known as *neurons* or *nodes*. A prototype vector is associated with each neuron and the map attempts to represent the data structure with optimal accuracy using a restricted set of neurons. At the same time the neurons are ordered on the grid so that similar prototypes are associated to the neighbouring neurons and dissimilar prototypes to neurons far from each other. The SOM learning algorithm achieves two important steps: (a) it clusters the input data into neurons and (b) spatial order, in the sense that similar input patterns

tend to produce response in neighbouring neurons. In this paper, the SOM is used to visualize the intrinsic structure of the multispectral data of the satellite image, and get an unsupervised classification of the corresponding pixels.

To be able to compare the ISOCLUST with the SOM results we set the SOM architecture to N=24 and N=10 neurons respectively, assuming that each neuron represent a cluster. The net inputs were the 7 bands described above. The SOM algorithm organizes the resulting maps as a 4x6 and a 2x5 bidimensional exagonal tasseled grids respectively. The resulting maps with N=24 and N=10 neurons are shown in Figure 4a and 4d respectively. Green hexagons represent the neurons and their size the number of pixels accumulated in each of them.

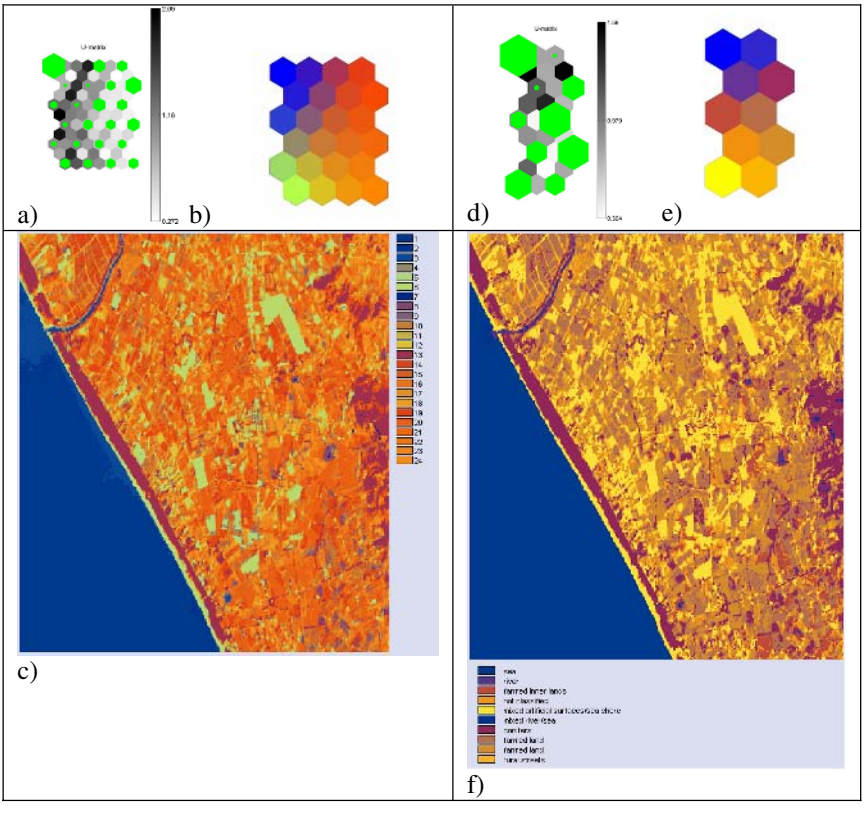


Fig. 4. a) A SOM map with 24 neurons, the green hexagons are the neurons. b) Map resulting from the Colouring Procedure described in the text. c) Original scene coloured according to the SOM colour map in panel b). Panels a), b) and c) are repeated in panels d), e), and f) for 10 neurons.

The different grey levels of the remaining hexagons exemplify the distance between each pair of neurons (i.e. the Euclidean distance between two prototypes), where greater distances correspond to darker hexagons.

Exploiting the SOM similarity propriety [10] a Colouring Procedure based on the Sammon's Projection [13] of the prototypes in a coloured plane was performed. The obtained projection allows to assign similar colours to similar neuron-prototypes on the SOM map. Each neuron was coloured according to the resulting projection of the prototype in the coloured map moving from blue to red for the left upper and the right lower corner respectively. The results are displayed in Figure 4b, and 4e for N=24 and N=10 neurons. The effects of the colouring procedures on the original scene are shown in Figure 4c and 4f for N=24 and N=10 neurons. respectively. The colouring process seems to cluster the maps into a reduced number of classes assigning similar colours to similar classes.

The colouring makes the human interpretation of the land cover easier. Comparing the results in Figure 4c with the prior knowledge of the scene we can see that the water (either of the sea or the river) appears in three different blue tones. Indeed these pixels fall in the three neurons in the upper left part of the map (panel 3b). Agricultural lands all over the image appear with different brown tones and the corresponding pixels fall in close neurons on the SOM map (panel 3b).

The yellow groups together the sea shore, the greenhouses and the urban area, while the magenta groups together the pine wood near the seashore and the mountain wood. In Figure 4c it is also possible to identify the rural streets, as for example at the border of the fields close to the river and two straight lines corresponding to the railway and to a major highway crossing the area. A less detailed but clearer clusterization comes from colouring the original scene with the SOM colour map obtained using 10 neurons as described in Figure 4d, 4e, and 4f.

It can be seen that the principal characteristics of the land cover, detectable in the 24 neurons' classification has been preserved in the 10 neurons' classification, and the clusters merged are those coloured with similar colour tones in the 24 classes case. This merge resulted useful for the interpretation of the original scene since similar structures were grouped together in the same cluster. The small number of clusters allows labeling them in a more precise way. For example, different types of conifers (i.e. seaside pinewood, conifers near some streets, and mountain woods) were all grouped in the same cluster. Three different kinds of farmed lands were attributed to three different clusters. Artificial surfaces (greenhouses and buildings) and seashore are displayed in only one single cluster. The river and the sea give rise to two different clusters. Even though some mixed classes are still present, the pure ones are more numerous than in the corresponding ISOCLUST experiment.

5 Conclusions

As shown above, the SOM based approach produces a result that makes the human interpretation of land covers through LANDSAT image easier , since it exploits the topological properties of the data in colouring the classes. In such a way, the interpreter, as in the classical interpretation of colour composite images, can achieve a colour driven labelling of unknown areas with a certain confidence. This property doesn't apply to the ISOCLUST results (see Figure 3 above), leaving to the interpreter the job of assigning the proper significance to each cluster.

In conclusion, both classifiers, even though based on an unsupervised algorithm, have proven to make a good job on the image under study, by revealing many intricate details of the landscape.

Acknowledgements

This work was partially funded by Centro Regionale di Competenza per l'Innovazione Applicata ai Beni Culturali ed Ambientali. The authors are thankful to the Professor Angela Pontrandolfo, dean of the department of Beni Culturali (Salerno University) and the dott. Alfonso Santoriello for the valuable information on the Paestum area.

References

1. Fu, K.S., Landgrebe, D.A., Phillips, T.L.: Information Processing of Remotely Sensed Agricultural Data. Proceeding of IEEE, Vol. 57 (1969) 639-653
2. Goldberg, M., Shlien, S.: A Clustering Scheme for Multispectral Images. IEEE Trans. SMC, Vol. 8 (1978) 86-92
3. Benediktsson, J., Philip, H. S., And Okan K. E.:Neural Network Approaches Versus Statistical Methods in Classification of Multi-source Remote Sensing Data. IEEE Transactions on Geoscience and Remote Sensing, Vol. 28:4 (1990) 540-551
4. McClelland, G.E., Dewitt, R.N., Hemmer, T.H., Matheson, L.N. Moe, G.O.: Multispectral Image-Processing with Three-Layer Back-Propagation Network. Proceedings of the International Joint Conference on Neural Networks, New York: I.E.E.E., Vol. 1 (1989) 151-153
5. Kamata. S., Kawaguchi, E.: A Neural Classifier for Multi-Temporal Landsat Images Using Spatial and Spectral Information. Proceedings of International Joint Conference on Neural Networks, Nagoya, Japan (1993) 2199-2202
6. Wang, Y., Jamshidi, M., Neville, P., Bales, C., Morain, S.: Multispectral Landsat Image Classification Using a Data Clustering Algorithm. Proceedings of 3th Intenational Conference on Machine Learning and Cybernetics, Shangai (2004) 4380-4384
7. Vassilas, N., Charou, E.: A New Methodology for Efficient Classification of Multispectral Satellite Images Using Neural Networks Techniques. Neural Processing Letters, Vol. 9 (1999) 35-43
8. Richards, J.A., Jia X.: Remote Sensing Digital Image Analysis. Springer-Verlag (1999) 58-59, 225,235-236
9. Chavez, P.S. Jr.: Image-Based Atmospheric Corrections Revisited and Revised. Photogrammetric Engineering and Remote Sensing Vol.62(9) (1996) 1025-1036
10. Kohonen, T.: Self-Organizing Maps. Series in Information Sciences, Vol. 30, Springer Verlag, Second Ed. (1997)
11. Mather, P. M.: Computer Processing of Remotely Sensed Images an Introduction. Second Ed., John Wiley Press (1999)
12. Büttner, G., et al: The European Corine Land Cover Project. XXth Congress of International Society for Photogrammetry and Remote Sensing, Istanbul , Turkey (2004)
13. Borg, I., and P. Groenen, P. Modern Multidimensional Scaling. Springer-Verlag, New York (1997).

Vehicle Detection and Tracking for Traffic Monitoring

Gian Luca Foresti and Lauro Snidaro

Department of Mathematics and Computer Science,
University of Udine, Via delle Scienze 206, 33100 Udine, Italy
{foresti, snidaro}@dimi.uniud.it

Abstract. This paper addresses some of the indications of the European Union for road safety by proposing a real-time traffic monitoring system for vehicle detection and tracking in bad illuminated scenarios. Several urban and extra-urban roads during the night or tunnels are characterized by low illumination, light spots, shadows, light reflections, etc. The main objectives of the proposed system are: (a) to monitor the traffic flow, (b) to estimate the vehicle's speed or determine the state of the traffic, (c) to detect anomalous situations, e.g. rising alarms in case of road accidents or stopped cars. Experimental results on real image sequences demonstrate the effectiveness of the proposed system.

1 Introduction

Currently, more than 40.000 persons are killed every year on EU roads and less than 1000 in the other modes of transport. The short term strategic objective of the Community is to halve the number of fatalities by 2010. The medium term objective for MISS is to cut by around 50% the number of persons killed or severely injured by 2010 [9].

The Monitor Integrated Safety System (MISS) project, funded by the European Sixth Framework Program, aims to develop an innovative platform to dynamically sense and predict natural and infrastructure conditions, so to improve safety and efficiency of transport operations in a multi-environmental scenario. This project wants to increase citizens and operators safety by enabling a just in time intelligent computation of an open dynamic road surveillance network and streamlining alerting tasks under the daily duty provided by clerical staff.

Improved safety can be achieved by assisting the human operator with a surveillance system providing enhanced sensory information about the environment [1]. In addition, the systems that control the flow of traffic can likewise be enhanced by providing sensory information regarding the current conditions in the environment.

Image processing and computer vision techniques are extensively used in traffic monitoring systems in order to increase safety [2-9]. For instance, the possibility to extract complex and high-level traffic information such as congestion or accidents allows to efficiently plan a path through the road network, to quickly bring assistance where needed or to deviate the traffic flow. In order to extract this type of information it is first necessary to separate moving objects from the scene. In this way, vehicles can be counted, and their trajectories and speeds estimated. Moreover, by using these parameters, it is possible to make a traffic classification between safe, congested or dangerous situations.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1198 – 1205, 2005.

One of the major difficulties of automatic monitoring traffic scenes is the variety of light conditions of outdoor scenes. D'Agostino [4] discussed the potentials of a commercial machine vision system for traffic monitoring and control. The basic requirements are low cost and robust performance, which have not been fully met till now. Moreover, problems of shadows and nighttime operation have not been solved. For example, the system cannot classify vehicles during nighttime. Vehicle speed was estimated by setting up two inspection zones. The system needs expensive special image processing hardware.

A background updating method for road traffic scenes is discussed in [5]. This system extracts vehicles using a spatio differentiation method to remove the adverse effects of vehicle shadows. The background updating method is based on the change ratio between the road surface brightness of the current input image and that of the old background image. In addition, the system must determine if vehicle(s) appearing in two successive frames are the same. The problem of estimating vehicle speed was not addressed.

In other systems, vehicle shapes are modeled using complex models [6], which cannot be processed in real-time with low-cost hardware. Yuan et al. [7] extract a vehicle and estimate its length, width, height and the number of units of the vehicle from a single perspective image captured by a camera placed at the roadside. The ultimate goal of their system is to classify vehicles into many categories, therefore reducing the gap between the requirement and the availability. However, their approach encounters the general problems in image segmentation, and the methods to identify the roof, side and front of a vehicle are quite ad hoc. Zhu et al. [8] presented a new approach to automatic traffic monitoring using 2D spatio-temporal images. A TV camera is placed above the highway to monitor the traffic through two slice windows for each traffic lane. One slice window is a detection line perpendicular to the lane and the other is a tracking line along the lane. The problems of vehicle counting, speed estimation and vehicle classification have been addressed. In particular, the vehicle classification is performed by using 3D measurements (length, width and height). The system, which has been called VISATRAM (vision system for automatic traffic monitoring), has been tested with real road images under various light conditions, including shadows in daytime and lights at night.

This paper presents a visual-based traffic monitoring system for analyzing vehicles' activities in bad illuminated roads or tunnels, where not only low light conditions, but also light spots, shadows and reflections characterize the observed environments.

2 System Architecture

The general system architecture is shown in Fig.1. Video signals are pre-processed to achieve 2D motion detection of mobile vehicles. To this end, a layered background subtraction approach [5] is used. A change detection module (CD) [5] is applied to compare each frame I(x,y) of the input image sequence with a background image BCK(x,y) which represents the monitored scene without moving objects.

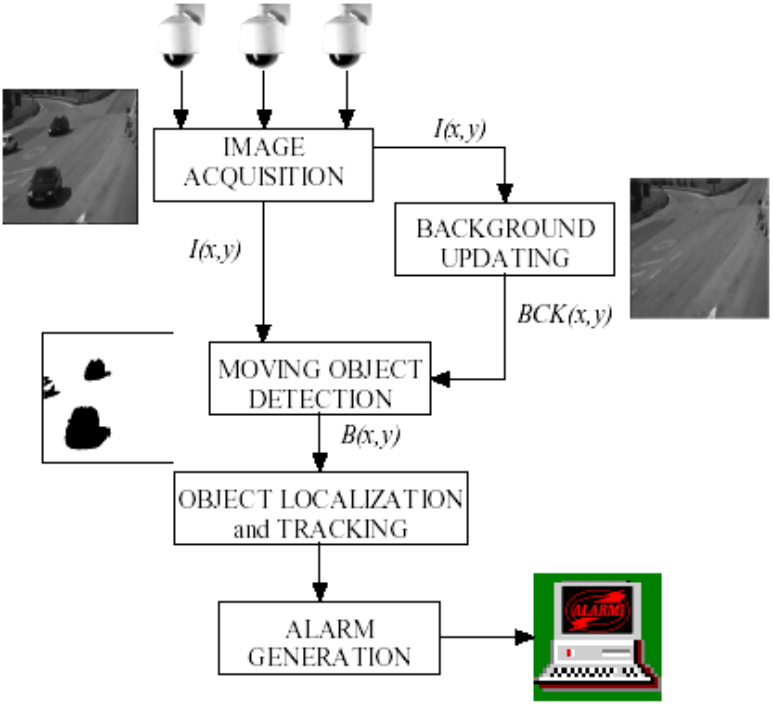


Fig. 1. General system architecture

This procedure generates a binary image *B(x,y)* where each pixel can assume two possible states: a static or a moving point. Groups of connected pixels (*blobs*) belonging to the class of moving points represent possible vehicles in the scene. A background updating procedure based on the Kalman Filter is applied to estimate significant illumination changes on the background scene *BCK(x,y)* [5].

The object localization module transforms 2D blob positions into 3D object positions referred to a top view map of the scene. An off-line camera calibration procedure is applied to compute for each camera the transformation matrix (obtained during the initial setup of the sensor). In order to solve the ill-posed problem of 2D into 3D transformation, the ground plane hypothesis is applied.

The object tracking module associates a potential vehicle detected at frame *t-1* to one of detected objects identified at frame *t*. This operation is very complex if performed in environments like tunnels or urban roads at night characterized by bad illumination conditions (e.g., low illumination, light spots, shadows, light reflections, etc.). In these conditions, classical object detection methods fail.

Finally, the obtained data are processed to count the number of vehicles moving in the scene, to estimate their speed and to determine the state of traffic. A man-machine interface is used to display in the most appropriate way both statistical information and alarms to a remote operator.

3 Traffic Monitoring

The main innovative part of the system is represented by the vehicle tracking module. It is based on different object features: (a) the 2D image position *M(x,y)* of the blob's center of mass computed at the time *t* and estimated at time *t-1*, and (b) the ratio $R = \frac{h}{w}$ between the height and the width of the minimum bounding rectangle (MBR) of the detected blob.

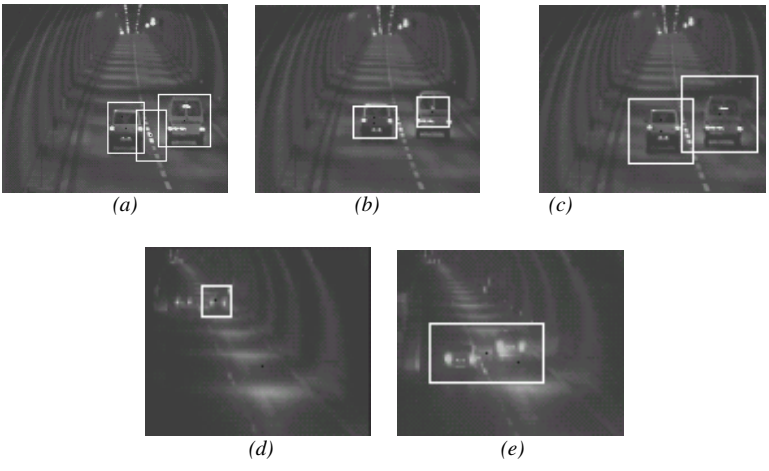


Fig. 2. Some problems of vehicle tracking in a badly illuminated tunnel: (a) false MBRs are detected, (b,c) the size of the MBRs can vary between successive frames, (d) true MBRs that disappear, (e) wrong (fused) MBRs can be detected

When a blob has been detected by the object detection module, the above features are computed in each frame of the sequence, and a matching function is applied to find true correspondences. Two constraints have been introduced in order to reduce the number of possible matches: (i) maximum displacement of the object over two consecutive frames (this constraint allows to define a confidence circle in which to search for candidates for association), and (b) maximum variation of the ratio *R* between two consecutive frames (this allows to consider only objects with a similar size). If the illumination conditions of the scene are good, the tracking problem reduces to match each detected object in the current frame with a candidate object of the previous frame.

Unfortunately, several urban roads and some tunnels are badly illuminated. Figure 2 shows the most common problems that can occur during the tracking operation in these conditions : (a) false MBRs can be detected (Fig. 3a), (b-c) the size of the MBRs can vary significantly between consecutive frames due to light spots (Figs. 3b and 3c), (d) true MBRs can disappear, (e) wrong MBRs can appear as consequence of erroneous blob detection due to partial occlusions, spot light, shadows, etc. In order to

allow the tracking operation also in complex situations, a Kalman filter is applied to estimate the position and the size of the MBR of each detected blob in the entire image sequence. Estimated MBRs can be compared with detected ones in order to eliminate false blobs, to take into account missed blobs due to occlusions, and to correct MBR' size and/or position.

3.1 Tracking Model

The following quantities of interest (QOIs) have been selected: (a) (x_1,y_1) - coordinates of the top-right corner of the MBR, (b) (x_2,y_2) - coordinates of the bottom-left corner of the MBR, (c) (x_c,y_c) - coordinates of the center of mass of the MBR, (c) (vx_1,vy_1), (vx_2,vy_2) and (vx_c,vy_c) - translation of the selected points along the x and y axes, respectively, (d) ρ – scale factor due to the perspective projection introduced by the camera. The state equations (position and speed) of the KF become:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{v}_k + \eta_k \tag{1a}$$

$$\mathbf{v}_{k+1} = \mathbf{C}\mathbf{v}_k + \omega_k \tag{1b}$$

where η_k and ω_k represent the process errors, the matrix **A** is the identity **I**. The matrix $\mathbf{C}=\alpha\mathbf{A}$, where α represents the (constant) deceleration factor, and

$$B = \begin{pmatrix} \tau & 0 & \tau(x_1 - x_2) & 0 & 0 & 0 & 0 & 0 \\ 0 & \tau & \tau(y_1 - y_2) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau & 0 & \tau(x_2 - x_c) & 0 & 0 \\ 0 & 0 & 0 & 0 & \tau & \tau(y_2 - y_c) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tau \end{pmatrix} \tag{2}$$

where τ is the time interval between two consecutive frames. The measurement equation of the KF becomes:

$$\mathbf{z}_{k+1} = \mathbf{H}\mathbf{v}_k + \hat{\mathbf{x}}_k + \gamma_k \tag{3}$$

where γ_k is the measurement error of the position vector and **H** is the observation matrix, i.e.,

$$\mathbf{H} = \begin{pmatrix} \tau & 0 & \tau(x^-_{1,k} - x^-_{c,k}) \\ 0 & \tau & \tau(y^-_{1,k} - y^-_{c,k}) \end{pmatrix} \tag{4}$$

We assume that η_k , ω_k and γ_k are independent random variables with zero means and covariance matrices, $\boldsymbol{Q_1}$, $\boldsymbol{Q_2}$ and $\boldsymbol{R}$. Finally, the equation of the Kalman filter for the position vector becomes:

(a) Prediction phase

$$\hat{\mathbf{x}}^-_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\hat{\mathbf{v}}_k \tag{5a}$$

$$\mathbf{P}^-_{k+1} = \mathbf{P}_k + \mathbf{Q} \tag{5b}$$

where $\boldsymbol{P}_k$ is the covariance matrix of the error estimation.

(b) Updating phase

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T \left(\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R} \right)^{-1} \tag{6a}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^- \right) \tag{6b}$$

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H} \right) \mathbf{P}_k^- \tag{6c}$$

The KF requires an initialization step. In particular, the initial values (at the time instant *t=0*) of the state vectors, $\mathbf{x_0}$ and $\mathbf{v_0}$, and of the covariance matrix $\boldsymbol{P_0}$ must be defined.

4 Experimental Results

Black/white cameras with near infrared response have been employed for experiments. Image grabbing was performed at 256x256 pixels resolution. The cameras have been placed to monitor the entire area of a bad illuminated road tunnel. Figure 3 shows an image sequence containing multiple moving vehicles.

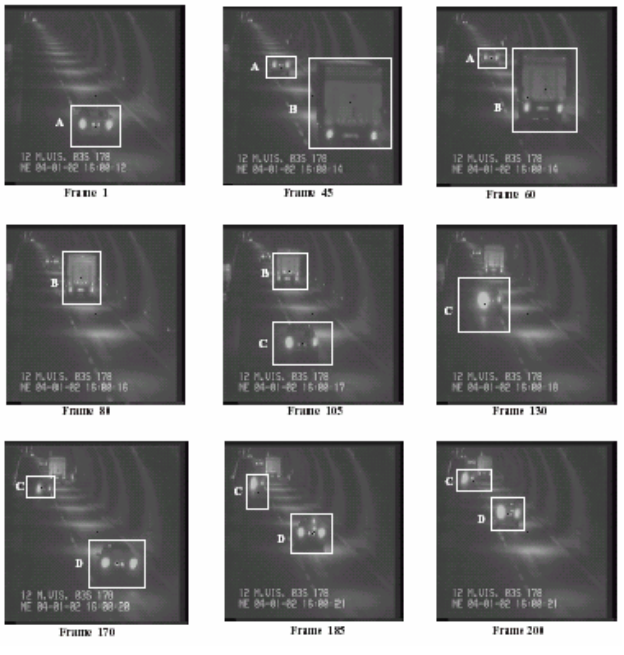


Fig. 3. Some frames of a long image sequence containing multiple moving vehicles

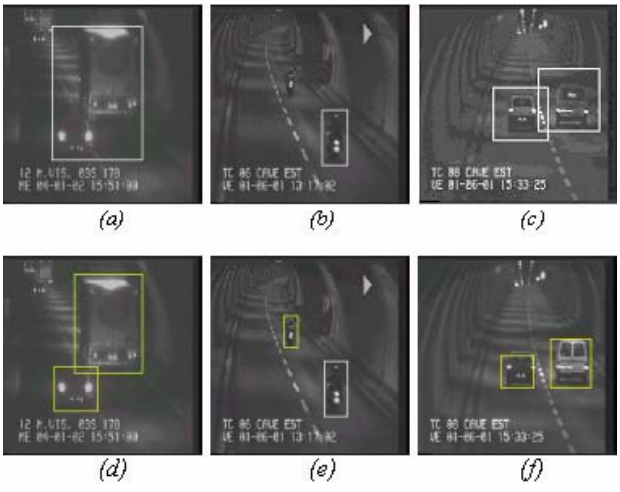


Fig. 4. Results of the proposed system in different situations

The images in Figure 3 belong to a 4 hours sequence (composed of about $4x10^5$ frames) characterized by low illumination, light spots and light reflection produced by the tunnel walls and by the rear lights of the vehicles. Figure 4 shows the results of the proposed system in three different complex situations: (1) fused MBRs due to blobs generated by closed vehicles (e.g., a vehicle overtakes another vehicle) (Fig. 4a), (2) missed MBRs due to occlusions or wrong blob detection (Fig. 4b), (3) wrong MBRs due to enlarged blobs generated by shadows or by light spots (Fig. 4c). The output of the system is shown in Fig. 4d, 4e and 4f, respectively.

It is worth noting that a percentage of about 9% of False Alarms (FAs) and a percentage of about 6% of Missed Alarms (MAs) have been obtained by the system on the 4 hours test sequence in Figure 4.

5 Conclusions

Within the European MISS project aimed at improving road safety, a system for real-time traffic monitoring in bad illumination conditions has been presented here. The system takes into account illumination changes and it performs vehicle tracking by means of an extended Kalman filter with a 2D state vector whose components represent each vehicle's position and speed. An affine model taking into account translations and scale changes is used to represent accurately vehicles' motion.

Acknowledgements

This work was supported by the European FP6 Project MISS "Monitor Integrated Safety System" (TST4-CT-2004-516235). Authors wish to thank Mr. Christian Nadalutti for his implementing support.

References

[1] G. L. Foresti, C.S. Regazzoni, and P.K. Varshney, *Multisensor Surveillance Systems:The Fusion Perspective,* Kluwer Academic Publishers, 2003.

[2] A.E. Grace, D. Pycock, H.T. Tillotson, and M.S. Snaith, "Active shape from stereo for highway inspection", *Machine Vision and Applications*, vol. 12, no. 1, pp.7-15, 2000.

[3] M. Aoky, "Imaging and analysis of traffic scene", *In.l Conf. on Image Processing*, Kobe, Japan, October 1999, pp. 28-32.

[4] S. D'Agostino, "Commercial machine vision system for traffic monitoring and control", SPIE, Vol. 1615, pp.180-186,1991.

[5] G.L.Foresti, "Real-time detection of multiple moving objects in complex image sequences", *Int. Journal of Imaging Systems and Technology*, Vol. 10, 1999, pp. 305-317.

[6] D. Bullock and S. Mantri, "Multimedia data model for video detection research", *Journal of Transportation Engineering*, vol. 121, no. 5, pp. 385-390, 1995.

[7] X. Yuan, Y-J. Lu and S. Sarraf, "Computer vision system for automatic vehicle classification", *Journal of Transportation Engineering*, vol. 120, no. 6, pp. 861-876, 1994.

[8] Z. Zhu, G. Xu, B. Yang, D. Shi, and X. Lin, "VISATRAM: a real-time vision system for automatic traffic monitoring", *Image and Vision Computing*, vol. 18, no. 10, pp.781-794, 2000.

[9] Online White Paper: "European transport policy for 2010: time to decide": http://europa.eu.int/comm/energy_transport/en/lb_en.html, 2001.

Consistent Labeling for Multi-camera Object Tracking*

Simone Calderara[1], Andrea Prati[2], Roberto Vezzani[1], and Rita Cucchiara[1]

[1] Dipartimento di Ingegneria dell'Informazione - University of Modena and Reggio Emilia - Via Vignolese, 905 - 41100 Modena, Italy
[2] Dipartimento di Scienze e Metodi dell'Ingegneria - University of Modena and Reggio Emilia - Via Allegri, 13 - 42100 Reggio Emilia, Italy

Abstract. In this paper, we present a new approach to multi-camera object tracking based on the consistent labeling. An automatic and reliable procedure allows to obtain the homographic transformation between two overlapped views, without any manual calibration of the cameras. Object's positions are matched by using the homography when the object is firstly detected in one of the two views. The approach has been tested also in the case of simultaneous transitions and in the case in which people are detected as a group during the transition. Promising results are reported over a real setup of overlapped cameras.

1 Introduction

Crowed environments such as metro stations, city markets, or public parks, are very difficult to monitor, also for human operators. Moreover, the recent development in video acquisition hardware and the consequent relaxation of its price have made possible a broad deployment of hundreds of cameras. The availability of multiple points of view and the redundancy of information allow a more precise (though not unfailing) monitoring of complex scenes.

Despite of the complexity increase, multiple camera systems exhibit the undoubt advantages of covering wide areas and enhancing the managing of occlusions (by exploiting the different viewpoints). However, the automatic merge of the knowledge extracted from single cameras are still challenging tasks, especially in application of *distributed people tracking*. The goal is to track multiple people moving in an environment observed by multiple cameras, tightly connected, synchronized and with partially overlapped views.

The solution to this problem must deal with two sub-problems: a reliable tracking in each camera system and the preservation of the identity of the people moving from a camera's view to the one of another camera. For this second problem, often know as *consistent labeling*, many solutions have been addressed. Among these, geometrical approaches require camera calibration. In outdoor environments with many cameras, placed in high positions over poles at unknown

* This work was supported by the project L.A.I.C.A. (Laboratorio di Ambient Intelligence per una Città Amica), funded by the Regione Emilia-Romagna, Italy.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 1206–1214, 2005.

distance, manual calibration could be difficult and time consuming to achieve. Thus, automatic camera calibration techniques have been proposed.

This paper focus on finding a reliable solution to the consistent labeling problem, also in challenging conditions such as the simultaneous presence of more people passing from a camera's field of view to the one of another camera. We exploit and improve a proposal of Khan-Shah [1] of edge of field of view (EOFOV) computation. The EOFOV lines, i.e. the boundaries between the field of views of partially overlapped cameras, are automatically created with a learning phase using a training video with a person walking along the limits of the field of view.

The paper presents a general model with a set of N overlapped cameras, and an improved and generalized technique for EOFOV learning. Then, the consistency between the extracted lines is exploited to compute a precise homography, used to establish the consistent labeling. The paper reports extensive experimental work in which very complex situations of multiple people crossing simultaneously the border of the FOV are considered. Experiments have been provided in a real setup with pairs of partially overlapped cameras monitoring an outdoor environment.

2 Related Works

Approaches to multicamera tracking can be generally classified into three main categories: geometry-based, color-based, and hybrid approaches. The former exploits geometrical relations and constraints between the different views to perform the consistent labeling process. Instead, *color-based* approaches base the matching essentially on the color of the tracks. For example, in [2] a color space invariant to illumination changes is proposed and histogram-based information at low (texture) and mid (regions and blobs) level are adopted. Conversely, the work in [3] uses stereoscopic vision to match tracks, but when this matching is not sufficiently reliable, color histograms are used to solve ambiguities. Finally, *Hybrid* approaches, belonging to the third class, mix information about the geometry and the calibration with those provided by the visual appearance. These last methods are based on probabilistic information fusion [4] or on Bayesian Belief Networks (BBN) [5][6], and sometimes a learning phase is required [7].

Geometry-based approaches can be further subdivided into calibrated and uncalibrated approaches. In [8], each camera processes the scene and obtains a set of tracks. Then, regions along the epipolar lines in pairs of cameras are matched and the mid-points of the matched segments are back-projected in 3D and then, with an homography, onto the ground plane to identify possible positions of the person within a probability distribution map (filtered with a Gaussian kernel). The probability distribution maps are then combined using outlier-rejection techniques to yield a robust estimate of the 2D position of the objects, which is then used to track them. A particularly interesting paper is reported in [9] in which homography is exploited to solve occlusions. Single camera processing is based on particle filter and on probabilistic tracking based on appearance to detect occlusions. Once an occlusion is detected, homography

is used to estimate the track position in the occluded view, by using the last valid positions of the track in it and the current position of the track in the other view (properly warped in the occluded one by means of the transformation matrix). A very relevant example of the uncalibrated approaches is the work of Khan and Shah [1]. Their approach is based on the computation of the so-called *Edges of Field of View*, i.e. the lines delimiting the field of view of each camera and, thus, defining the overlapped regions. Through a learning procedure in which a single track moves from one view to another, an automatic procedure computes these edges that are then exploited to keep consistent labels on the objects when they pass from one camera to the adjacent.

Our approach is a suitable modification of this proposal to compute, starting from the EOFOV lines extraction, the homography relation between the two ground plane in an automatic way. This transformation is adopted for the consistent labeling problem.

3 The Edge of Field of View

The proposed approach belongs to the class of uncalibrated geometry-based techniques and it is a suitable modification of the proposal reported by Khan and Shah in [1]. The basic idea relies on learning the calibration parameters by means of the creation of the so-called *edges of field of view* (EOFOV).

Let us suppose that the system is composed of a set $\mathbf{C} = \{C^1, C^2, ..., C^n\}$ of n cameras, with each camera C^i being overlapped with at least another camera C^j. Projecting the limits of the field of view (FOV) of a camera C^i on the ground plane ($z = 0$), the so-called *3D FOV lines* [1] can be obtained. In particular, they correspond to the intersection between the ground plane and the rectangular pyramid with its vertex at the camera optical center (the camera view frustum). A 3D FOV line is denoted by $L^{i,s}$, where s indicates the equation of the line in the 2D coordinates of the camera C^i that generates the 3D FOV line. In particular, the four 3D FOV lines $L^{i,s_h} \,|\, h = 1 \ldots 4$ (where s_h corresponds to the image borders $x = 0$, $x = x_{max}$, $y = 0$, and $y = y_{max}$) can be computed. A projection of a 3D FOV line of camera C^i may be visible in another camera C^j partially overlapped with C^i. The FOV line (in 2D) of the line s of camera C^i seen by the camera C^j will be then denoted with $L^{i,s}_j$ and represents one of the EOFOV lines for the camera C^j. Each line $L^{i,s}_j$ divides the image on camera C^j into two half-planes, one overlapped with camera C^i and the other disjoint. The intersection of the overlapped semi-planes defined by the EOFOV lines from camera C^i to camera C^j defines the overlapping area Z^i_j. In Fig. 1 the placement of four cameras in our campus is depicted. Fig. 2(a) shows a view of cameras C^1 and C^2 and the EOFOV $L^{1,s}_2$ is depicted with s correspondent to the image border $x = 0$ of C^1.

The EOFOV lines are created with a training procedure; the process is iterated for each pair (C^i, C^j) of partially overlapped cameras. To this aim, we need the correspondences of a certain number of points on the ground plane in the two considered views. Thus, as proposed in [1], during the training phase a

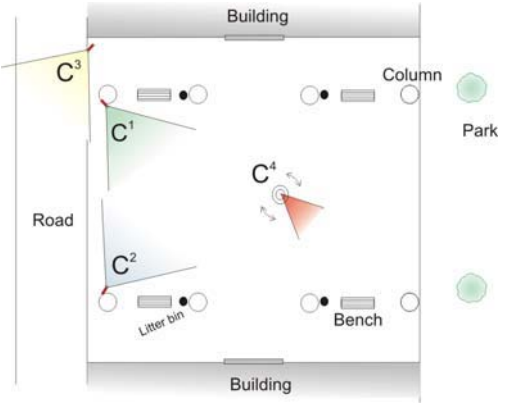


Fig. 1. Map of our real setup

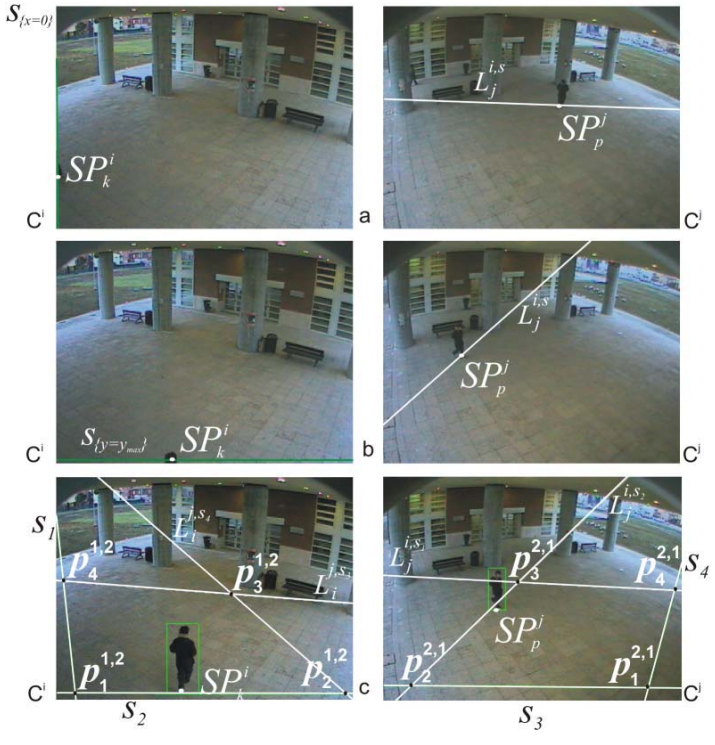


Fig. 2. Examples of EOFOV creation: (a) correct creation, (b) incorrect creation (using Khan-Shah approach), and (c) the proposed solution.

single person moves freely in the scene, with the minimum requirements to pass through at least two points of each limit of the FOV of two overlapped cameras.

Let us call O_k^i the object segmented and tracked with label k in the camera C^i and SP_k^i the point of contact with the ground plane (*support point*, hereinafter).

The support point can be easily computed as the middle point of the bottom of the bounding box of the blob.

Given the constraint to have a single moving person in the training video, problems of consistent labeling do not occur. Thus, when the object is detected in the C^j camera too, and tracked with the p label, it is directly associated to O_k^i. Therefore, in this moment (known as the moment of "camera handoff") the support point SP_k^i can be directly associated to SP_p^j (if it is visible). In this case the point SP_k^i lies on the EOFOV line $L_i^{j,s}$ for the camera C^i. The equation of each line $L_i^{j,s}$ is computed by collecting a set of coordinates of the support point SP_k^i detected at the camera handoff and exploiting a Least Square optimization (Fig. 2(a) shows SP_k^i and SP_p^j at the camera handoff instant).

In the method proposed in [1], the points SP are extracted at the camera handoff moment. This can bring to false correspondences, as in the case of a person entering from the bottom of the image (Fig. 2(b)). In such a situation, the head in C^i is in correspondence with the feet in C^j. However, this matching is reliable enough if the goal is only the consistent labeling at the camera handoff instant (as in [1]) and if the persons have the same height. Instead, if an exact correspondence is required, for example to compute an homography transformation, we must verify that the matching points belong to the same real point (e.g., the feet).

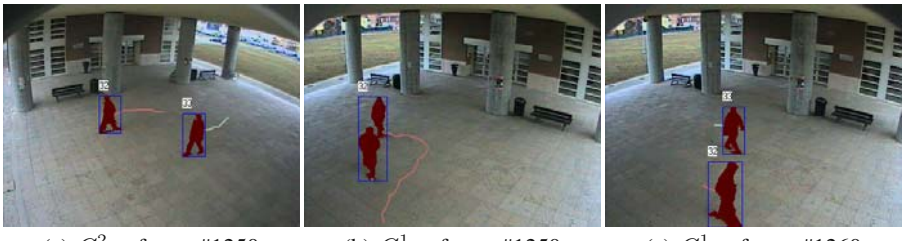
To this aim, we modified the approach by delaying the computation of the EOFOV lines to the moment in which the object is completely entered the scene of the new camera (see Fig. 2(c)). This can bring to a displacement of the line with respect to the actual limit of the image, but it assures the correct match of the feet's position in the two views. The amount of the displacement depends on the position of the feet w.r.t. the image limit. As a consequence, the actual FOV lines s are neither coincident nor parallel to the image border. In Fig. 2(c) the lines $(L_j^{i,s_1}, L_j^{i,s_2})$, $(L_i^{j,s_3}, L_i^{j,s_4})$ correspondent to (s_1, s_2) in C^i, (s_3, s_4) in C^j respectively are depicted.

4 Consistent Labeling Approach

The approach proposed in [1] establishes the consistent labeling only in the exact moment of the camera handoff from C^i to C^j. This technique assign the object O_p^j in the camera C^j to the object in the camera C^i with the minimum distance from the EOFOV $L_i^{j,s}$ corresponding to the side s from which the object enters. This approach has two main limits: if two or more objects cross simultaneously (Fig. 3), then an incorrect labeling could be established; if two or more objects are merged from the view of C^j at the moment of the camera handoff, but then they separate, the consistent labeling with the corresponding labels of C^i can not be recovered.

This last limit is evident in Fig. 4: camera C^2 sees the two objects (O_{32}^2 and O_{33}^2) as separated (Fig. 4(a)), but they are merged in a single object when they appear in camera C^1 and only the label 32 is assigned to it(Fig. 4(b)).

Fig. 3. Example of simultaneous crossing of two objects



(a) C^2 at frame #1250 (b) C^1 at frame #1250 (c) C^1 at frame #1260

Fig. 4. Example of simultaneous crossing of two objects

We propose to overcome these problems by means of homography, thus extending the matching search to the whole zone of overlap of field of view. For two overlapped cameras C^i and C^j, the training procedure computes the overlapping areas Z_j^i and Z_i^j. The four corners of each of these areas defines a set of four points, $P_j^i = \{p_1^{i,j}, p_2^{i,j}, p_3^{i,j}, p_4^{i,j}\}$ and $P_i^j = \{p_1^{j,i}, p_2^{j,i}, p_3^{j,i}, p_4^{j,i}\}$, where the subscripts indicate corresponding points in the two cameras (see Fig. 2(c)). These four associations between points of the camera C^i and points of the camera C^j on the same plane $z = 0$ are sufficient to compute the homography matrix H_j^i from camera C^i to camera C^j. Obviously, the matrix H_i^j can be easily obtained with the equation $H_i^j = (H_j^i)^{-1}$.

Each time a new object O_k^i is detected in the camera C^i in the overlapping area (not only at the moment of the camera handoff), its support point SP_k^i is projected in C^j by means of the homographic transformation. Calling $(x_{SP_k^i}, y_{SP_k^i})$ the coordinates of the support point SP_k^i, we can write the projected point in homogeneous coordinates:

$$[a, b, c]^T = H_j^i \left[x_{SP_k^i}, y_{SP_k^i}, 1 \right] \tag{1}$$

The projected point $\widetilde{SP_k^j}$ corresponds on the image plane of C^j to the projective coordinates $\widetilde{x^j} = \frac{a}{c}$ and $\widetilde{y^j} = \frac{b}{c}$. These coordinates could not correspond to the support point of an actual object. For the match with O_k^i we select the object in

C^j whose support point is at the minimum distance in the 2D plane from these coordinates:

$$O_k^i \longleftrightarrow O_p^j \mid p = \arg\min_q \; D(\widetilde{SP_k^j}, SP_q^j) \;\; \forall q \in \mathbf{O^j} \tag{2}$$

where $D(\cdot)$ denotes the Euclidean distance and $\mathbf{O^j}$ is the set of objects detected in C^j. The results achieved with this approach in the two cases above reported are shown in Fig. 3 and Fig. 4, respectively, where the correct label assignment is achieved.

5 Experimental Results

To test the system, we have installed four partially overlapped cameras in our department (see Fig. 2 for two snapshots and Fig. 1 for a map). The tests were carried out using a single camera probabilistic and appearance based tracking module [10]. EOFOV lines of the two cameras have been computed over a training video of 8000 frames. As an evidence of the goodness of the automatically obtained homography we report in Fig. 5 the mosaic image of two frames obtained merging a frame of a camera with a homographically distorted frame of the other camera.

To test the consistent labeling algorithm, instead, we have tested the system not only in the simple conditions of the training phase, but also in presence of simultaneous transitions of more than one person at a time (Sync. Trans.) and in presence of transitions in which two people are merged (Merged Trans.) in a single track during the camera handoff and split far from the EOFOV.

In Table 1 we have reported the obtained results; the number of camera transitions correctly identified (in which the consistent labeling is verified) and the number of wrong correspondences are reported in the last two columns of

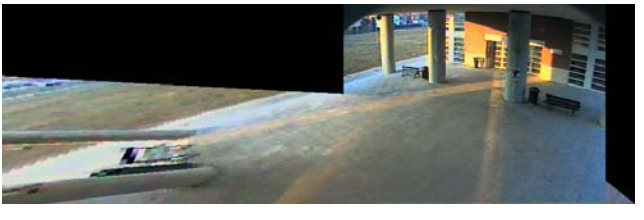
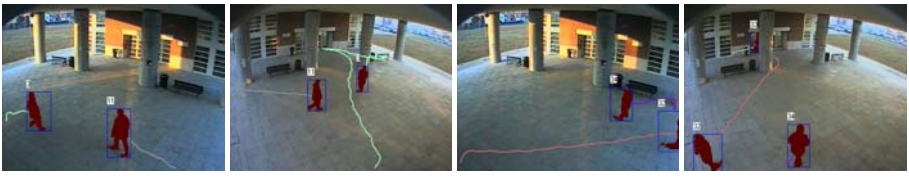


Fig. 5. Automatically obtained mosaic image through homography

Table 1. Experimental results

	Sync. Trans.	Merged Trans.	N frames	N transitions	Correct	Incorrect
Video 1	No	No	8500	41	39	2
Video 2	No	No	3000	5	5	0
Video 3	Yes	No	1800	14	13	1
Video 4	Yes	Yes	2000	7	6	1
Video 5	Yes	Yes	500	2	2	0



(a) C^1 at frame #776 (b) C^2 at frame #776 (c) C^1 at frame #1490 (d) C^2 at frame #1490

Fig. 6. Some snapshots of the output of the system after consistent labeling

the table. It is evident that the system has a very high accuracy. The incorrect matches are mainly due to errors in the lower modules, i.e. in the segmentation and the tracking algorithms. Some snapshots of the output of the system after the consistent labeling assignment are reported in Fig. 6.

6 Conclusions

This paper presents a new method for establishing consistent labeling in a multi-camera system. Its main contributions can be summarized as follows:

1. the improvement of the automatic calibration procedure proposed in [1] to overcome to known problems; this procedure is based on the computation of the edges of field of view (EOFOV) lines;
2. the computation of the homography matrices between two overlapped views by using the EOFOV lines;
3. the exploitation of the homographic transformation to establish consistent labeling in the whole overlapping area, in order to recover the correct labels in the case of objects that enter as merged and then split.

The reported experiments demonstrate the accuracy of the proposed method, also in difficult situations.

References

1. Khan, S., Shah, M.: Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. IEEE Trans. on PAMI 25 (2003) 13551360
2. Li, J., Chua, C., Ho, Y.: Color based multiple people tracking. In: Proc. of IEEE Intl Conf. on Control, Automation, Robotics and Vision. Volume 1. (2002) 309314
3. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S.: Multi-camera multiperson tracking for easyliving. In: Proc. of IEEE Intl Workshop on Visual Surveillance. (2000) 310
4. Kang, J., Cohen, I., Medioni, G.: Continuous tracking within and across camera streams. In: Proc. of IEEE Intl Conference on Computer Vision and Pattern Recognition. Volume 1. (2003) I267 I272
5. Chang, S., Gong, T.H.: Tracking multiple people with a multi-camera system. In: Proc. of IEEE Workshop on Multi-Object Tracking. (2001) 1926
6. Dockstader, S., Tekalp, A.: Multiple camera tracking of interacting and occluded human motion. Proc. of the IEEE 89 (2001) 14411455

7. Chang, T., Gong, S., Ong, E.: Tracking multiple people under occlusion using multiple cameras. In: Proc. of British Machine Vision Conf. Volume 2. (2000) 566576
8. Mittal, A., Davis, L.: Unified multi-camera detection and tracking using region-matching. In: Proc. of IEEE Workshop on Multi-Object Tracking. (2001) 310
9. Yue, Z., Zhou, S., Chellappa, R.: Robust two-camera tracking using homography. In: Proc. of IEEE Intl Conf. on Acoustics, Speech, and Signal Processing. Volume 3. (2004) 14
10. Cucchiara, R., Grana, C., Tardini, G.: Track-based and object-based occlusion for people tracking refinement in indoor surveillance. In: Proc. of ACM 2nd InternationalWorkshop on Video Surveillance & Sensor Networks. (2004) 8187

Author Index