

ADVANCES IN *SOFT COMPUTING* 45

Marek Kurzynski  
Edward Puchala  
Michal Wozniak  
Andrzej Zolnierek (Eds.)

# Computer Recognition Systems 2

 Springer



# Advances in Soft Computing

## Editor-in-Chief

Prof. Janusz Kacprzyk  
Systems Research Institute  
Polish Academy of Sciences  
ul. Newelska 6  
01-447 Warsaw  
Poland  
E-mail: kacprzyk@ibspan.waw.pl

---

Further volumes of this series can be found on our homepage: [springer.com](http://springer.com)

Barbara Dunin-Keplicz, Andrzej Jankowski, Andrzej Skowron, Marcin Szczuka (Eds.)  
*Monitoring, Security, and Rescue Techniques in Multiagent Systems*, 2005  
ISBN 978-3-540-23245-2

Frank Hoffmann, Mario Köppen, Frank Klawonn, Rajkumar Roy (Eds.)  
*Soft Computing Methodologies and Applications*, 2005  
ISBN 978-3-540-25726-4

Mieczyslaw A. Kłopotek, Sławomir T. Wierzchon, Krzysztof Trojanowski (Eds.)  
*Intelligent Information Processing and Web Mining*, 2005  
ISBN 978-3-540-25056-2

Abraham Ajith, Bernard de Bactis, Mario Köppen, Bertram Nickolay (Eds.)  
*Applied Soft Computing Technologies: The Challenge of Complexity*, 2006  
ISBN 978-3-540-31649-7

Mieczyslaw A. Kłopotek, Sławomir T. Wierzchon, Krzysztof Trojanowski (Eds.)  
*Intelligent Information Processing and Web Mining*, 2006  
ISBN 978-3-540-33520-7

Ashutosh Tiwari, Joshua Knowles, Erel Avineri, Keshav Dahal, Rajkumar Roy (Eds.)  
*Applications and Soft Computing*, 2006  
ISBN 978-3-540-29123-7

Bernd Reusch, (Ed.)  
*Computational Intelligence, Theory and Applications*, 2006  
ISBN 978-3-540-34780-4

Miguel López-Díaz, María ç. Gil, Przemysław Grzegorzewski, Olgierd Hryniewicz, Jonathan Lawry  
*Soft Methodology and Random Information Systems*, 2006  
ISBN 978-3-540-34776-7

Ashraf Saad, Erel Avineri, Keshav Dahal, Muhammad Sarfraz, Rajkumar Roy (Eds.)  
*Soft Computing in Industrial Applications*, 2007  
ISBN 978-3-540-70704-2

Bing-Yuan Cao (Ed.)  
*Fuzzy Information and Engineering*, 2007  
ISBN 978-3-540-71440-8

Patricia Melin, Oscar Castillo, Eduardo Gómez Ramírez, Janusz Kacprzyk, Witold Pedrycz (Eds.)  
*Analysis and Design of Intelligent Systems Using Soft Computing Techniques*, 2007  
ISBN 978-3-540-72431-5

Oscar Castillo, Patricia Melin, Oscar Montiel Ross, Roberto Sepúlveda Cruz, Witold Pedrycz, Janusz Kacprzyk (Eds.)  
*Theoretical Advances and Applications of Fuzzy Logic and Soft Computing*, 2007  
ISBN 978-3-540-72433-9

Katarzyna M. Węgrzyn-Wolska, Piotr S. Szczepaniak (Eds.)  
*Advances in Intelligent Web Mastering*, 2007  
ISBN 978-3-540-72574-9

Emilio Corchado, Juan M. Corchado, Ajith Abraham (Eds.)  
*Innovations in Hybrid Intelligent Systems*, 2007  
ISBN 978-3-540-74971-4

Marek Kurzynski, Edward Puchala, Michał Wozniak, Andrzej Zolnierz (Eds.)  
*Computer Recognition Systems 2*, 2007  
ISBN 978-3-540-75174-8

Marek Kurzynski, Edward Puchala,  
Michal Wozniak, Andrzej Zolnierek (Eds.)

---

# Computer Recognition Systems 2

 Springer

## Editors

Prof. Marek Kurzynski  
Dr. Edward Puchala  
Dr. Michal Wozniak  
Dr. Andrzej Zolnierek

Wroclaw University of Technology  
Faculty of Electronics  
Wybrzeze Wyspianskiego 27  
50-370 Wroclaw  
Poland

Library of Congress Control Number: 2007935490

ISSN print edition: 1615-3871

ISSN electronic edition: 1860-0794

ISBN-13 978-3-540-75174-8 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007

Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: by the authors and SPS using a Springer L<sup>A</sup>T<sub>E</sub>X macro package

Printed on acid-free paper SPIN: 12070022 89/SPS 5 4 3 2 1 0

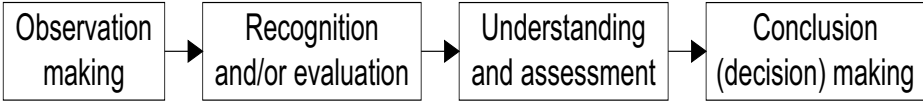
---

# Preface

## From Recognition Through Understanding to Decision Making

More than 60 years have passed since the time when first attempts to explain the mechanisms of information processing in human (animals') brain were made. We associate them with the first physical model of neuron proposed by W.S. McCulloch and W. Pitts (1947) which some years later (1957) was used by F. Rosenblatt as a basis of the concept of artificial neural networks. However, in the after years the expectations that the basic mental processes can be effectively explained by their reduction to simple logical interactions between artificial neurons evidently failed. The relationships between neurophysiology and computer science occurred more complicated. On one hand, a progress in neurophysiology and in psycho-neurology resulted in better understanding of natural thinking mechanisms and indirectly stimulated efforts in creation of more effective computer-based information processing and decision making methods. On the other hand, progress in design of computer-based information processing, including experimental data processing methods and systems influenced the bio- and medical sciences in cerebral processes investigation. This, in particular, concerns modern anatomic and functional brain imaging methods based on the CT, NMR, PET and/or SPECT modalities which in the last decades became a powerful tool of achieving new experimental data in neurophysiology. Due to them we can try to learn how the human brain works and to use this knowledge to improve the artificial information processing methods and systems by simulation of the natural processes. Recognition of observed objects (visual, speech, textual etc. data, physical, bio- or geophysical phenomena, medical symptoms, etc., generally called *patterns*) is an important element of a cognitive process consisting, in general, of several stages shown in the following scheme (Fig. 1):

It consists in assigning to a given object (result of observation) the name of a previously defined similarity class. Classification (definition of similarity



**Fig. 1.** Main stages of a cognitive process

classes in a given observation space) of objects and their recognition are thus two complementary tasks. Moreover, recognition in most cases is not a simple, one-step operation. In image processing it may be preceded by a rough delimitation of regions where objects of particular interest are expected to occur. Objects recognition may also include, as its first step, a preliminary detection of objects' features or evaluation of objects' parameters. The last not only may be used as a basis of final object recognition but also it may lead to the recognized objects more detailed characterization. Therefore, recognition and evaluation of observed objects can be considered as a complex of mutually related operations.

An observed object being recognized may evoke a next series of questions, e.g.: what are (or might be) the relations between it and other objects in its environment, what follows from the occurrence of the recognized object in the given place and time, to what degree occurrence of this object is important for the environment the observations have been acquired, etc. Giving reasonable replies to this type of questions is possible as a result of understanding and assessment of the recognized object. However, the terms *reasonable* and *understanding* have no exact and commonly accepted interpretations. Roughly speaking, *reasonable* means here *based on commonly in pure sciences accepted data analysis, inference and decision making methodology*. This definition seems to be wide enough to accept a large spectrum of logical and/or statistical methods. The meaning of understanding is much more controversial. J. R. Searle has shown (1980) that understanding needing interpretation of textual information can be, in fact, realized in the form of a sequence of mechanical operations performed on simple textual data.

On the contrary to it, R. Penrose has stated (1994) that *conscience* is a necessary condition of *understanding* and, thus, computers are unable to understand anything they do or are dealing with. However, it is not necessary to resolve this, philosophical rather, contradiction in order to reach a practical goal: creation of effective methods of reasonably answering certain types of questions concerning the recognized objects. The problem is thus as follows: how to represent the knowledge about a given area of interest in a form acceptable by computers in order to make them able to reasonably reply the given class of questions. The questions can be inspired by the interests of the users and may be aimed at reaching some research, educational or practical goals.

The papers included into this book concern both, the pattern recognition principles and their practical applications. However, this classification is to a certain degree arbitrary, the concepts of novel methods very often being illustrated by practical examples and some pattern recognition applications including original modifications of the corresponding algorithms. In the, above-mentioned,

first group of papers design and evaluation of effective pattern recognition algorithms is a dominating problem. The methods based on Support Vector Machines (SVM), mathematical statistics, formal grammars, fuzzy sets, decision trees, etc. are presented. The problems and methods of input data acquisition, selection, image enhancement and segmentation also have been presented in a group of papers. Among the pattern recognition applications the problems of computer-aided medical diagnosis are the most frequently presented. Early cancer detection, analysis of ECG, EEG and AMG signals, stem cells and bio-molecular structures analysis, 3D human organs modeling, etc. are in the center of interest of authors of numerous papers. A large group of papers also concerns speech and handwritten characters recognition and textual data analysis. The list of other pattern recognition applications presented in the book is also large. The road traffic analysis, assessment and prediction of situation of enterprises, personal identification based on human face analysis, etc. as selected examples can be mentioned. This short résumé of the book contents shows that pattern recognition is a vital and very fruitful from the point of view of practical applications research area.

The perspectives of further development of pattern recognition are closely connected with a progress in computer-aided observed data understanding and assessment methods. Let us remark that on the lower levels of recognition the elements of domain knowledge are provided in the form of adjusted parameters of recognizing algorithms, of weights of inter-neuronal connections in artificial neural networks or of the reference data sets in learning systems. In higher-level recognition systems using more advanced domain knowledge representation methods are necessary. Among them various forms of statistical or functional models, domain ontologies, formal grammars, etc. can be mentioned. Each of the above-mentioned forms leads, respectively, to a specific approach to advanced recognition and interpretation of observed objects. It seems that a fusion of knowledge representation methods, data mining and logical inference based on incomplete or uncertain data methods will chart the main pattern recognition directions of development in the future.

Juliusz L. Kulikowski



---

# Contents

---

## Part I: Image Processing and Computer Vision

---

<b>Strategies of Shape and Color Fusions for Content Based Image Retrieval</b> <i>Paweł Forczmański, Dariusz Frejlichowski</i> . . . . .	3
<b>Determining the Contour of Cylindrical Biological Objects Using the Directional Field</b> <i>Robert Koprowski, Zygmunt Wrobel</i> . . . . .	11
<b>Trajectory Fusion for Multiple Camera Tracking</b> <i>Ariel Amato, Murad Al Haj, Mikhail Mozerov, Jordi González</i> . . . . .	19
<b>Object Selection Based on Clustering and Border Objects</b> <i>J. Arturo Olvera-López, J. Ariel Carrasco-Ochoa, J. Francisco Martínez-Trinidad</i> . . . . .	27
<b>Multi-directional Multi-resolution Transforms for Zoom-Endoscopy Image Classification</b> <i>Roland Kwitt, Andreas Uhl</i> . . . . .	35
<b>Image Retrieval for Image Theft Detection</b> <i>Ondřej Horáček, Jakub Bican, Jan Kamenický, Jan Flusser</i> . . . . .	44
<b>Practical Evaluation of the Basic Concepts for Face Localization</b> <i>Maciej Smiatacz</i> . . . . .	52
<b>Hybrid Fuzzy Clustering Method</b> <i>Tomasz Przybyła</i> . . . . .	60
<b>Structural Image Analysis Based on Ontological Models</b> <i>Juliusz L. Kulikowski</i> . . . . .	68

<b>Homogeneity of Pixel Neighborhoods in Gray Level Images Investigated by the Grade Correspondence Analysis</b> <i>Maria Grzegorek</i> .....	76
<b>Comparison of Hierarchical Cluster Analysis Methods Applied to Image Segmentation by Watershed Merging</b> <i>Jakub Smolka, Maria Skublewska-Paszowska</i> .....	84
<b>Modified Diffusion to Image Denoising</b> <i>Dariusz Borkowski</i> .....	92
<b>New Results in Generation of Accurate 3D Multiview Representation of Monotonous Polyhedrons</b> <i>Wojciech S. Mokrzycki, Andrzej Salamończyk</i> .....	100
<b>On the Application of Distance Transformation in Digital Image Colorization</b> <i>Przemysław Lagodzinski, Bogdan Smolka</i> .....	108
<b>Local Analysis of Stereo Image Pairs with Polynomial Series</b> <i>Przemysław Głomb</i> .....	117
<b>The Architecture of the Face and Eyes Detection System Based on Cascade Classifiers</b> <i>Andrzej Kasiński, Adam Schmidt</i> .....	124
<b>Morphological Edge Detection Algorithm and Its Hardware Implementation</b> <i>Marek Kraft, Andrzej Kasiński</i> .....	132
<b>Numerical Complexity Reduction in Live-Wire Algorithm</b> <i>Wojciech Więclawek, Ewa Piętka</i> .....	140
<b>Image Segmentation Using Adaptive Potential Active Contours</b> <i>Arkadiusz Tomczyk</i> .....	148
<b>Towards an Exploration of GCA Ordered Pixels</b> <i>Maria Grzegorek</i> .....	156
<b>A Simple Method of Multiple Camera Calibration for the Joint Top View Projection</b> <i>Mikhail Mozerov, Ariel Amato, Murad Al Haj, Jordi González</i> .....	164
<b>Spiral Scanning of Faces to Obtain Complete 3D View Representation of Monotonous Polyhedra</b> <i>Wojciech S. Mokrzycki, Andrzej Salamończyk</i> .....	171

<b>Face Detection in Color Images Using Primitive Shape Features</b>	
<i>Murad Al Haj, Ariel Amato, Xavi Roca, Jordi González</i> . . . . .	179
<b>Detection of Near-Duplicated Image Regions</b>	
<i>Babak Mahdian, Stanislav Saic</i> . . . . .	187
<b>Automatic Merging of 3D Attribute Meshes</b>	
<i>Krzysztof Skabek, Agnieszka Tomaka</i> . . . . .	196
<hr/>	
<b>Part II: Features, Learning and Classifiers</b>	
<hr/>	
<b>Learning Machines Information Distribution System with Example Applications</b>	
<i>Norbert Jankowski, Krzysztof Grąbczewski</i> . . . . .	205
<b>Maximum Margin Classification on Convex Euclidean Metric Spaces</b>	
<i>André Stuhlsatz, Hans-Günter Meier, Andreas Wendemuth</i> . . . . .	216
<b>Optimization of the SVM Screening Kernel– Application to Hit Definition in Compound Screening</b>	
<i>Karol Kozak, Marta Kozak, Katarzyna Stapor</i> . . . . .	224
<b>Estimation of the Preference Relation on the Basis of Medians from Pairwise Comparisons in the Form of Difference of Ranks</b>	
<i>Leszek Klukowski</i> . . . . .	232
<b>Feature Selection for High-Dimensional Data – A Pearson Redundancy Based Filter</b>	
<i>Jacek Biesiada, Włodzisław Duch</i> . . . . .	242
<b>Some Approximation Algorithms for Minimum Vertex Cover in a Hypergraph</b>	
<i>Barbara Maźbic-Kulma, Krzysztof Sęp</i> . . . . .	250
<b>Reference Set Reduction Algorithms Based on Double Sorting</b>	
<i>Marcin Raniszewski</i> . . . . .	258
<b>Using Graph Mining Methods in Searching for Frequent Patterns in Graph-Based Design Representation</b>	
<i>Barbara Strug</i> . . . . .	266

**Stability of Feature Points Defined by Approximating Quadric Surfaces**

*Leszek Luchowski* ..... 274

**Generalisation of a Language Sample for Grammatical Inference of GDPLL( $k$ ) Grammars**

*Janusz Jurek* ..... 282

**Pattern Classification Using Efficient Linear Classifiers with Small Number of Weights**

*Magdalena Topczewska* ..... 289

**Markov Chain Model for Tree-Based Genetic Algorithm Used in Computer Aided Design**

*Anna Paszyńska, Marian Jabłoński, Ewa Grabska* ..... 297

**Binary Shape Characterization Using Morphological Boundary Class Distribution Functions**

*Marcin Iwanowski* ..... 305

**Fuzzy-aided Parsing for Pattern Recognition**

*Marzena Bielecka, Marek Skomorowski* ..... 313

**On Speeding Up the Learning Process of Neuro-fuzzy Ensembles Generated by the Adaboost Algorithm**

*Marcin Korytkowski, Leszek Rutkowski, Rafał Scherer* ..... 319

**Active Hypercontour as Information Fusion Method**

*Piotr S. Szczepaniak* ..... 327

**Effective Ad Recognition Using Schur-type Signal Parametrization**

*Paweł Biernacki* ..... 334

**Feature Extraction of Gray-Scale Handwritten Characters Using Gabor Filters and Zernike Moments**

*Ryszard S. Choraś* ..... 340

**Pattern Discovery Through Separable Data Projections**

*Leon Bobrowski, Volodymir Mashtalir, Magdalena Topczewska* ..... 348

**Support Vector Machine Classifier with Feature Extraction Stage as an Efficient Tool for Atrial Fibrillation Detection Improvement**

*Paweł Kostka, Ewaryst Tkacz* ..... 356

<b>Classification Accuracy in Local Optimal Strategy of Multistage Recognition with Fuzzy Data</b> <i>Robert Burduk</i> .....	364
<b>Experiments on Classifiers Obtained Via Decision Tree Induction Methods with Different Attribute Acquisition Cost Limit</b> <i>Wojciech Penar, Michal Wozniak</i> .....	371
<b>Fast Adaptive Fourier Transform for Fourier Descriptor Based Contour Classification</b> <i>Dariusz Puchala, Mykhaylo Yatsymirskyy</i> .....	378
<b>Separable Decomposition of Graph Using <math>\alpha</math>-cliques</b> <i>Henryk Potrzebowski, Jarosław Stańczak, Krzysztof Sep</i> .....	386
<b>A Feature Selection Approach in Problems with a Great Number of Features</b> <i>Pawel Kosla</i> .....	394
<hr/>	
<b>Part III: Speech and Word Recognition</b>	
<hr/>	
<b>Normalisation of Confidence Voting Methods Applied to a Fast Handwritten OCR Classification</b> <i>Juan Ramón Rico-Juan, José M. Iñesta</i> .....	405
<b>Extraction of the Visual Features from the Audio-Visual Speech Signal and the Utilization of These Features for the Speaker Identification</b> <i>Josef Chaloupka</i> .....	413
<b>Different Approaches to Class-Based Language Models Using Word Segments</b> <i>Raquel Justo, M. Inés Torres</i> .....	421
<b>Recognition of Ultra Low Resolution Word Images Using HMMs</b> <i>Farshideh Einsele, Rolf Ingold, Jean Hennebert</i> .....	429
<b>Manageable Phrase-Based Statistical Machine Translation Models</b> <i>Ghada Badr, Eric Joanis, Samuel Larkin, Roland Kuhn</i> .....	437

<b>Automatic Detection of Disorders in a Continuous Speech with the Hidden Markov Models Approach</b> <i>Marek Wiśniewski, Wiesława Kuniszyk-Józkowiak, Elżbieta Smolka, Waldemar Suszyński</i> . . . . .	445
<b>The Polish Finger Alphabet Hand Postures Recognition Using Elastic Graph Matching</b> <i>Joanna Marnik</i> . . . . .	454
<b>Character Recognition Based on Fourier Transform and CDWT in Postal Applications</b> <i>Mirosław Maszewski, Mirosław Miciak</i> . . . . .	462
<b>Preprocessing for Real-Time Handwritten Character Recognition</b> <i>Bartosz Paszkowski, Wojciech Bieniecki, Szymon Grabowski</i> . . . . .	470
<b>Handwritten Word Recognition with Combined Classifier Based on Tri-grams</b> <i>Jerzy Sas, Andrzej Zolnierak</i> . . . . .	477
<b>Semi-automatic Handwritten Word Segmentation Based on Character Width Approximation Via Maximum Likelihood Method and Regression Model</b> <i>Jerzy Sas, Marek Kurzynski</i> . . . . .	485
<b>Sentence Boundary Verification in Polish Text</b> <i>Krzysztof Simiński</i> . . . . .	493
<hr/>	
<b>Part IV: Medical Applications</b>	
<hr/>	
<b>Synthesis of Static Medical Images – An Example of Melanocytic Skin Lesions</b> <i>Zdzisław S. Hippe, Lukasz Piątek</i> . . . . .	503
<b>Morphological Spectra as Tools for Texture Analysis</b> <i>Juliusz L. Kulikowski, Malgorzata Przytulska, Diana Wierzbicka</i> . . . . .	510
<b>Breast Contour Detection for the Aesthetic Evaluation of Breast Cancer Conservative Treatment</b> <i>Jaime S. Cardoso, Maria J. Cardoso</i> . . . . .	518
<b>Simple EEG Driven Mouse Cursor Movement</b> <i>Jan Kněžík, Martin Dražanský</i> . . . . .	526

<b>The Prediction of Fetal Outcome by Applying Neural Network for Evaluation of CTG Records</b> <i>Michał Jeżewski, Janusz Wróbel, Krzysztof Horoba, Adam Gacek, Norbert Henzel, Jacek Leski</i> .....	532
<b>Determining Weights of Symptoms in a Diagnostic Inference</b> <i>Ewa Straszcka</i> .....	542
<b>Hough Transform, (1+1) Search Strategy and Watershed Algorithm in Segmentation of Cytological Images</b> <i>Maciej Hrebień, Józef Korbicz, Andrzej Obuchowicz</i> .....	550
<b>Preliminary Investigations Regarding the Blood Volume Estimation in Pneumatically Controlled Ventricular Assist Device by Pattern Recognition</b> <i>D. Komorowski, M. Gawlikowski</i> .....	558
<b>‘Sonar’ – Region of Interest Identification and Segmentation Method for Cytological Breast Cancer Images</b> <i>Tomasz Nieczkowski, Andrzej Obuchowicz</i> .....	566
<b>Validation of Automatic ECG Processing Management in Adaptive Distributed Surveillance System</b> <i>Piotr Augustyniak</i> .....	574
<b>Estimation of a Muscle Force from a Mechanomyographic Signal During a Contraction of a Single Motor Unit</b> <i>Piotr Kaczmarek, Andrzej Kasiński</i> .....	581
<b>Automatic Segmentation of EMG Signals Based on Wavelet Representation</b> <i>Przemysław Mazurkiewicz</i> .....	589
<b>Elimination of Linear Structures as an Attempt to Improve the Specificity of Cancerous Mass Detection in Mammograms</b> <i>Marcin Bator, Leszek J. Chmielewski</i> .....	596
<b>Automatic Counting of Neural Stem Cells Growing in Cultures</b> <i>Anna Korzyńska</i> .....	604
<b>Value of Digital Image Analysis in Research and Diagnosis of Urine Bladder Cancer</b> <i>A. Dulewicz, D. Pietka, P. Jaszczak</i> .....	613

<b>Movement Tracking of Coronary Artery Segment in Angiographic Images Sequences by Template Matching Method</b> <i>Hanna Goszczyńska</i> .....	621
<b>Computer Ultrasonic Imaging of the Tongue Shape Changes in the Process of Articulation of Vowels</b> <i>Krzysztof J. Opielinski, Tadeusz Gudra, Joachim Migda</i> .....	629
<b>The Biometric Signals Processing</b> <i>Zbigniew Gomolka, Tomasz Lewandowski</i> .....	637
<b>Digital Dental Models and 3D Patient Photographs Registration for Orthodontic Documentation and Diagnostic Purposes</b> <i>Agnieszka Tomaka, Michal Tarnawski, Leszek Luchowski, Barbara Lisniewska-Machorowska</i> .....	645
<b>Gait Motor Disturbances in Neurological Diseases Diagnosis</b> <i>Jan Piecha</i> .....	653
<b>Analysis of Head-Mounted Wireless Camera Videos for Early Diagnosis of Autism</b> <i>Basilio Noris, Karim Benmachiche, Julien Meynet, Jean-Philippe Thiran, Aude G. Billard</i> .....	663
<b>Automatic Generation of 3D Lung Model</b> <i>Dominik Spinczyk, Ewa Piętka</i> .....	671
<b>Semi-automatic Seed Points Selection in Fuzzy Connectedness Approach to Image Segmentation</b> <i>Pawel Badura, Ewa Piętka</i> .....	679
<b>Control of Dexterous Hand Via Recognition of EMG Signals Using Combination of Decision-Tree and Sequential Classifier</b> <i>Andrzej Wolczowski, Marek Kurzynski</i> .....	687
<b>Posterior Cruciate Ligament – 3D Visualization</b> <i>Piotr Zarychta</i> .....	695

---

**Part V: Various Applications**

---

<b>A Self-training Approach for Automatically Labeling IP Traffic Traces</b> <i>Francesco Gargiulo, Claudio Mazzariello, Carlo Sansone</i> .....	705
---	-----



<b>A Case-Base Reasoning System for Predicting the Economic Situation of Enterprises – Tacit Knowledge Capture Process (Externalization)</b>	
<i>Jan Andreasik</i> .....	718
<b>A Structural Pattern Analysis Approach to Iris Recognition</b>	
<i>Hugo Proença</i> .....	731
<b>Enhancing Real-Time Human Detection Based on Histograms of Oriented Gradients</b>	
<i>Marco Pedersoli, Jordi González, Bhaskar Chakraborty, Juan J. Villanueva</i> .....	739
<b>Decisive Factors in the Annotation of Emotions for Spoken Dialogue Systems</b>	
<i>Zoraida Callejas, Ramón López-Cózar</i> .....	747
<b>A Hybrid Big Rock Detection Algorithm Based on Multiple Images Fusion and Watershed</b>	
<i>Mohammad T. Al Modarresi, Mohammad S. Tabatabaei, Mohammad T. Sadeghi</i> .....	755
<b>Design Description Hypergraph Language</b>	
<i>Ewa Grabska, Grażyna Ślusarczyk, Michał Glogaza</i> .....	763
<b>Articulation Rate Recognition by Using Artificial Neural Networks</b>	
<i>Izabela Szczurowska, Wiesława Kuniszzyk-Józkowiak, Elżbieta Smolka</i> .....	771
<b>Recognition of Cycles of Repeated Hypoxia on the Basis of Time Periods in Biological Model</b>	
<i>Beata Sokolowska, Adam Jozwik</i> .....	778
<b>Fingerprint Reference Point Detection Using Neighbourhood Influence Method</b>	
<i>Piotr Porwik, Lukasz Wieclaw</i> .....	786
<b>Gene Expression Programming-Based Method of Optimal Frequency Set Determination for Purpose of Analogue Circuits' Diagnosis</b>	
<i>P. Jantos, D. Grzechca, T. Golonek, J. Rutkowski</i> .....	794
<b>Prediction of Economic Situation of Small and Medium Enterprises Using Bayesian Network</b>	
<i>Barbara Kuczowska, Andrzej Burda, Zdzisław S. Hippe</i> .....	802

<b>Ensembles of Artificial Neural Networks for Predicting Economic Situation of Small and Medium Enterprises</b> <i>Andrzej Burda, Barbara Kuczumska, Zdzisław S. Hippe</i> .....	808
<b>The Performance of the Haar Cascade Classifiers Applied to the Face and Eyes Detection</b> <i>Adam Schmidt, Andrzej Kasiński</i> .....	816
<b>FPGA Implementation of the Gradient Adaptive Lattice Filter Structure for Feature Extraction</b> <i>Bogusław Szlachetko, Andrzej Lewandowski</i> .....	824
<b>Adaptive Selection of Feature Set Dimensionality for Classification of DNA Microarray Samples</b> <i>Henryk Maciejewski</i> .....	831
<b>Human Lips Recognition</b> <i>Michał Choraś</i> .....	838
<b>Model Based Vehicle Extraction and Tracking for Road Traffic Control</b> <i>Bartłomiej Płaczek, Marcin Staniek</i> .....	844
<b>Optimization of Linear Fuzzy Gene Network Model Searching</b> <i>A. Gintrowski, Ewaryst Tkacz, Paweł Kostka, A. Wieclawek, U. Mazurek</i> .....	852
<b>Author Index</b> .....	861

**Image Processing and Computer Vision**

---

# Strategies of Shape and Color Fusions for Content Based Image Retrieval

Paweł Forczmański and Dariusz Frejlichowski

Szczecin University of Technology  
{pforczmanski,dfrejlichowski}@wi.ps.pl

**Summary.** The aim of this paper is to discuss a fusion of the two most popular image features - color and shape - in the aspect of content-based image retrieval. It is clear that these representations have their own advantages and drawbacks. Our suggestion is to combine them to achieve better results in various areas, e.g. pattern recognition, object representation, image retrieval, by using optimal variants of particular descriptors (both, color and shape) and utilize them in the same time. To achieve such goal we propose two general strategies (sequential and parallel) for joining elementary queries. They are used to construct a system, where each image is being decomposed into regions, basing on shapes with some characteristic properties - color and its distribution. In the paper we provide an analysis of this proposition as well as the initial results of application in Content Based Image Retrieval problem. The original contribution of the presented work is related to the fusion of several shape and color descriptors and joining them into parallel or sequential structures giving considerable improvements in content-based image retrieval. The novelty is based on the fact that many existing methods (even complex ones) work in the same domain (shape or color), while the proposed approach joins features from different areas.

## 1 Introduction

Content-Based Image Retrieval (CBIR) has been a very attractive topic for many years in the scientific society. Although there are many academic solutions known (for example IBM's QBIC and MIT's Photobook), there are almost no commercial or industrial applications present. It is mostly caused by not trivial problems, developers of such systems should solve. The most important are the way the images are represented in the database and the way they are being compared. The automatic recognition of objects, which are placed in the image plane, can utilize various features. The most popular and widely used are: shape, texture, color, luminance, context of the information (background, geographical, meteorological etc.) and behavior (mostly movement). It is possible to use more than one feature at the same time, but such an approach is rather rare. Usually, each recognition method is limited to only one feature. On the other hand, the literature survey shows that combining images coming from different sources (instead of different

features of the same image) is gaining the popularity among researchers [5]. The image fusion has been widely accepted in diverse fields like medical imaging, aircraft navigation guidance, robotic vision, agricultural and satellite imaging. Image fusion is a necessary stage for these applications to achieve better understanding of the observed phenomena as well as improving decision making. The approach is motivated by weaknesses of individual imageries, which can be eliminated by joining their unique features. However, there are many situations when different sensors are not available and only one type of image is a source of features. Hence, we should use as much object information as it is possible. It is obvious that every type of object representation has its advantages and drawbacks, and the choice is not always easy and evident. It depends on many conditions, e.g. application and user requirements, situation during the image acquisition process (especially the hardware parameters, weather or lighting). In the paper we focus on visual descriptors related to shape and color, since they are the most popular in the literature and guarantee good efficiency when it comes to single-feature type of recognition [1, 2, 9, 13]. To improve the CBIR efficiency we join them into parallel or sequential structures.

## 2 Visual Descriptors

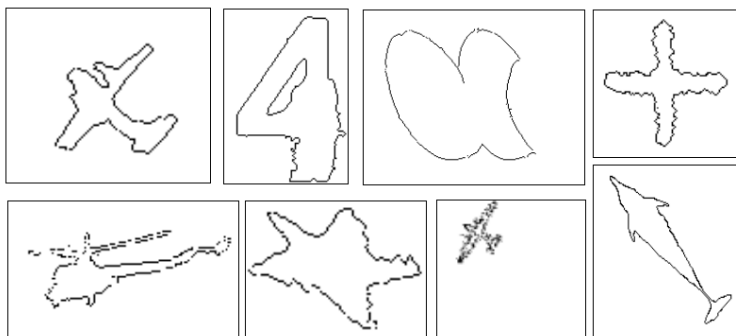
### 2.1 Shape Descriptors

The shape (silhouette, contour) is a very widely used object representation. It is very useful, when we are going to identify a particular object in the image, for example in medicine (e.g. cell shapes in microscopic images), optical character recognition and many other systems, where signs are being used, robotics (e.g. machine vision, searching for faults, machine positioning, orientation in distance), ecology (e.g. pollution monitoring), criminology (e.g. fingerprints), military application (e.g. aircrafts and tanks recognition, tracking, maps identifying), visual data retrieval. Shapes of different objects are usually easy to localize and distinguish. For example in a problem of object recognition in remote sensing, the extracted shape is more stable than other features, influenced by changes in illumination, color or contrast. The shape used for recognition can be considered as a binary object, which is represented by a whole, including its interior or as a boundary (contour). It is crucial to uniquely characterize the shape and stay invariant to translation, scale and rotation [8].

Shape descriptors can be classified in various ways [4, 8, 10, 12, 13]. The first classification is based on mentioned earlier distinction between object boundary and the whole shape. The second very popular manner (as described e.g. in [13]) is based on whether the shape is represented as a whole (global approaches) or by a set of primitives (structural methods). The third one distinguishes between spatial and transform domain [10].

Every shape descriptor should be resistant to as many shape distortions as possible. These distortions are considered as differences between object under recognition and the reference object belonging to the same class. In real recognition tasks one has to take into consideration the following problems divided

into three main groups. The first one includes spatial transformations of an object, mainly translation, rotation in the image plane, the change of its size and the influence of the projection into resultant two-dimensional shape. The second group covers distortions of the object itself: varying amount of points, noise, discontinuity and occlusion, which is equivalent to lack of some parts or added parts to shape. The third group of problems is related to the contour representation only and includes, among others, the selection of starting point and the direction of tracing the outline. Of course any other problem, which is typical to particular representation and is a consequence of using it, will belong to this group. The elements of the second group are the most challenging and difficult to solve. Few examples of real influences of above problem on objects boundaries are presented in Fig. 1.



**Fig. 1.** Few examples of boundaries affected by shape distortions after extracting them from real images: first row  $\tilde{U}$  an aircraft on airfield, a digit on car license plate, leaves, a symbol on a banknote; second row  $\tilde{U}$  a helicopter, a star on flag, an aircraft in the sky, a dolphin on sea

The most important problem with shape descriptors is the fact that usually when particular method is really 'brilliant' in presence of one or even few distortions, it completely fails in the presence of another one (ones).

## 2.2 Color Descriptors

Color is the second most important feature taken into consideration, when it comes to content-based image retrieval. The use of color is motivated by the way the Human Visual System (HVS) works. It is known that in good lighting conditions human-being pays attention: first to intensity and color of objects, second to shape and movement, then to texture and other properties. There have been many color descriptors proposed in the past, most of them based on different color-subspace histograms and dominant values. Nowadays, when the MPEG-7 standard [1] is being introduced, the most promising are compact descriptors, which join color information and its distribution: Scalable Color

(SCD), Dominant Color (DCD), Color Layout (CLD). In our recent works we have been also utilized specific simplified representations, like RGB color histogram (RGBHIST), intensity thumbnail (8x8 pixels) of an image (IBOX8), three (R,G,B) thumbnails (RGBBOX8), mean RGB value together with mean intensity (RGBI), dominant values H and V in HSV color model (DHV).

Descriptors presented above were successfully implemented in prototype software realizing CBIR tasks (querying similar images by example). Sample results of image query based on RGBHIST can be seen in Fig. 2. The most upper-left image is a query image, while the rest was provided by a system. Although all resulting images have similar color characteristics, it is clear, that using of only one feature is not optimal and is not in accordance with a way HVS works. Hence it is obvious that joining descriptors and decisions based on them will lead to the improvement in CBIR efficiency.



**Fig. 2.** Result of query involving color-only descriptors (RGBHIST)

### 3 Fusion of Queries

The specific of large visual data sets (consisting of several hundreds thousands of similar images) is associated with the high probability of the situation, when single descriptor used for query would give false hits. Hence, the problem of joining descriptors or decisions based on them is so important. The idea is not new and there are many examples of its successful implementation [5, 6, 7]. However, most solutions in that field use joint descriptors from the same domain (e.g. color or texture). Since the most real images contain objects that feature not only one kind of attributes i.e. color or shape, the proposed methods are not suitable for them. They are focused on either binary objects or textured rasters. The strategy of joining queries in different domains is, in fact, rather rare.

A typical system used for visual material querying and indexing consists of four main elements: the feature extractor, the comparison and classification

block, the storage sub-system, and the front-end. The efficiency of such system depends mainly on the performance of the comparisons and the accuracy of description. Our proposal, presented here, is to combine few queries (using several descriptors in the shape and color domains) at the same time in order to utilize the advantages of particular, single-feature methods while reducing the influence of their drawbacks.

The idea of combining few methods in the field of pattern recognition is not new. Fusion at decision level is employed to increase classification accuracy of an image beyond the level accomplished by individual classifiers. Rank-based decisions provide more opportunities compared to other numerical score measurements. Fusion at the level of features, on the other hand, is much simpler to implement, but suffers from the incompatibility of individual scales and requires applying universal classifiers (instead of feature-specific, which is undoubtedly better). For example, in the domain of shape recognition, the UNL-F descriptor is a combination of UNL and Fourier transforms [11]. In [10] two combined approaches were also explored, namely: moments and Fourier descriptors, and moments and UNL-F features. In [3] a combination of global (moments) and local (Structural Decomposition) was successfully proposed. These examples were given to show the recent tendency of combining various approaches to achieve better performance of resultant descriptor.

Our main idea, presented in this paper, is a presentation of two strategies of query fusion: sequential and parallel one using both color and shape descriptors. In this work, we propose a framework system that provides for both feature-types comparison and spatial query for unconstrained color images. The idea is based on the two-tier approach. First, each image is decomposed into regions (based on their shapes) which have properties, such as color and its distribution. Second, a color-based comparison is applied. In this way, these images are compared by comparing their individual objects. The system accommodates partial matching of objects that are the most important parts of an image not only using their shape, but also a color. The elementary query processes can be joint in sequential or parallel way.

The first approach (presented in Fig. 3) consists of  $n$  iterative queries. The first one uses  $D1$  descriptor which limits the initial dataset and creates a subset  $IS1$  by means of seeking and sorting images according to some similarity measure (e.g. distance metrics or correlation). Next, the  $IS1$  subset is used as a initial dataset for query with  $D2$ . The whole process is repeated  $n$  times giving the resulting images. In this approach it is important to create the sequence of descriptors in a way that each consecutive query produces successive approximation of the required result.

In the second approach (presented in Fig. 4) we assume parallel use of descriptors to get  $n$  results. After that we apply certain voting rules (i.e. two-out-of-three, three-out-of-five and similar) and select the resulting images. It should be noticed that in both approaches we can use different classifiers adequately to different features, which give distinctly better and more trustworthy results. Each strategy has its own advantages and drawbacks. The first one (sequential) utilizes an intuitive flow, which is similar to iterative way of seeking images by



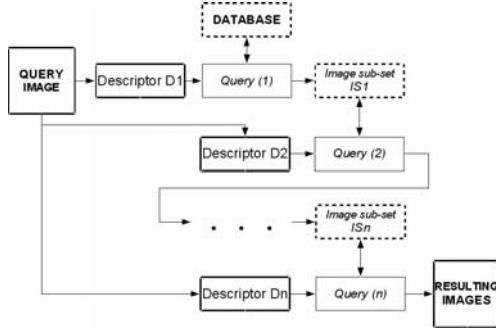


Fig. 3. Sequential structure of joining queries

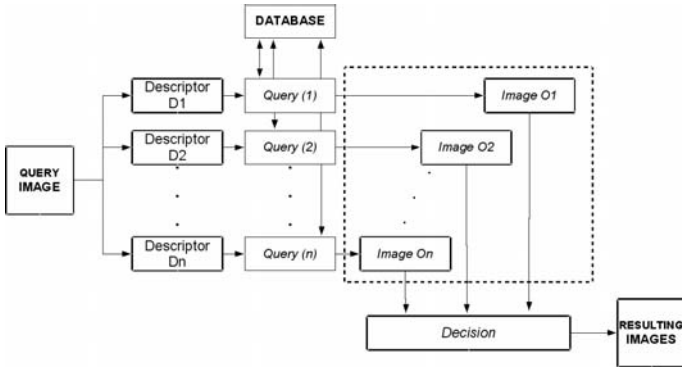


Fig. 4. Parallel structure of joining queries

humans, while the second one (parallel) makes it possible to avoid a situation, when images correctly found at the first stage are eliminated during the process of reducing the dataset at the further stages. In practice each strategy has its own specific application. According to several experiments we have performed, the sequential order is better for image retrieval based on examples, while the parallel one tends to be more appropriate for seeking images used for composing photo-mosaics, when only one resulting image is needed. Sample results of image query employing two stage, sequential seeking (based firstly on shape, then on RGB histogram) are presented in Fig. 5. As it can be seen, in comparison to the results presented in Fig. 2, the results has been distinctly improved.

The strategy of joining queries can be presented on an simplified example of a CBIR system which consists of three descriptors and three comparators, respectively. We denote the mean retrieval rate of the pairs descriptor-comparator as:  $P_1, P_2$  and  $P_3$ . We omit a problem of decision itself by taking each single retrieval as independent event and assuming that each pair works in its optimal conditions. More sophisticated approaches can be found in [6, 7]. The total rate



**Fig. 5.** Result of query involving combined features

$P$  of such a combined system can be calculated (on the interval  $(0, 1)$ ) according to the following formula:

$$P = P_1 P_2 P_3 + P_1 P_2 \bar{P}_3 + P_1 \bar{P}_2 P_3 + \bar{P}_1 P_2 P_3, \quad (1)$$

where:  $\bar{P}_1 = 1 - P_1$ ,  $\bar{P}_2 = 1 - P_2$ ,  $\bar{P}_3 = 1 - P_3$ .

For example, if the retrieval accuracy of a single descriptor-comparator pair is equal to 0.9 (90%), which is a typical rate for current methods, then the combined accuracy will increase to 0.972.

## 4 Summary

In the article we showed some aspects related to multi-tier content-based image retrieval, employing shape and color information. The ideas can be applied to many different fields of digital image processing and pattern recognition. They are universal and, as it was proved, can be successfully implemented. The main advantage over the existing methods is the possibility of joining descriptors from various domains and compare them using specific metrics to get better efficiency. Since the proposed structures are the elements of a framework, they can collect descriptors focused on different image classes, e.g. faces, road signs, logos, etc. However, we should remember that there are still some unsolved problems and questions regarding successful image retrieval, especially reducing the semantic gap between low-level features and high-level meaning.

## References

1. Bober M.: MPEG-7 Visual Shape Descriptors, IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 6 (2001) 716-719
2. Deng Y., Manjunath B. S., Kenney C., Moore M. S., Shin H.: An Efficient Color Representation for Image Retrieval, IEEE Transactions on Image Processing, vol. 10, no.1 (2001) 140-147

3. Foggia P., Sansone C., Tortorella F., Vento M.: Combining statistical and structural approaches for handwritten character description, *Image and Vision Computing*, vol. 17, no. 9 (1999) 701–711
4. Jain A. K.: *Fundamentals of Digital Image Processing*, Prentice Hall, 1989
5. Kukharev G., Mikłasz M.: Face Retrieval from Large Database, *Polish Journal of Environmental Studies*, vol. 15, no. 4C (2006) 111–114
6. Kuncheva L.I.: Combining classifiers: Soft computing solutions, in: *Pattern Recognition: From Classical to Modern Approaches*, World Scientific Publishing Co., Singapore (2001) 427–452
7. Kuncheva L.I.: A theoretical study on six classifier fusion strategies, *IEEE Transactions on PAMI*, 24, no. 2 (2002) 281–286
8. Loncaric S.: A survey on shape analysis techniques, *Pattern Recognition*, vol. 31, iss. 8 (1998) 983–1001
9. Manjunath B. S., Ohm J.-R., Vasudevan V. V., Yamada A.: Color and Texture Descriptors, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6 (2001) 703–715
10. Mehtre B. M., Kankanhalli M. S., Lee W. F.: Shape measures for content based image retrieval: a comparison, *Information Proc. & Management*, vol. 33 (1997) 319–337
11. Rauber T.W., Steiger-Garcia A.S.: 2-D form descriptors based on a normalized parametric polar transform (UNL transform), *Proc. MVA'92 IAPR Workshop on Machine Vision Applications* (1992)
12. Wood J.: Invariant pattern recognition: a review, *Pattern Recognition*, vol. 29, iss. 1 (1996) 1–17
13. Zhang D., Lu G.: Review of shape representation and description techniques, *Pattern Recognition*, vol. 37, iss. 1 (2004) 1–19

---

# Determining the Contour of Cylindrical Biological Objects Using the Directional Field

Robert Koprowski and Zygmunt Wrobel

University of Silesia, Institute of Computer Science, 41-200 Sosnowiec,  
Bedzinska 39, Poland  
{koprow,wrobel}@us.edu.pl

**Summary.** In the paper we present the employment of the direction field for contour detection and the segmentation of cylindrical objects, especially an eye iris. We propose a method of calculating the direction field for this kind of objects and present the obtained results. An analysis of the possibility of extending this approach on objects of different kinds is also included.

## 1 Introduction

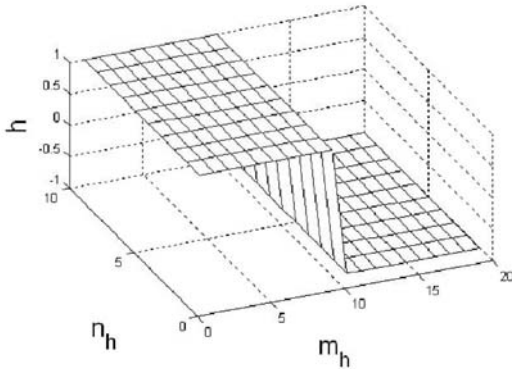
Image segmentation is one of the largest branches of image processing. The quality of obtained results is still improving and the segmentation algorithms are more and more universal. One of areas on interest is the methods of the segmentation of medical images, which require the most sophisticated algorithms due to the individual variation of patients and, consequently, low repeatability of acquired images. Many recent publications suggest dealing with this problem using known methods but extending them with new elements. For instance, image segmentation can be supplemented with contour detection methods, e.g. the snake method or its modification with the vector field added [21]. There are also used morphological methods [17], methods employing Hough transformation [5], [7], [18], local gradients [12], [16] texture analysis [15] and wavelet [11] or beamlets method [3], [4]. These methods give satisfying results in selected, very specific applications [20], [23]. Unfortunately, the mentioned methods are just one of many steps in detection and approximation of contour of cylindrical objects, which is documented, for instance in [1]. On the other hand, there are very successful algorithms based on traditional method of thresholding with modifications, e.g. calculating the energetic function [24]. Quite good results are also obtained in the recognition of cylindrical objects - e.g. the face tracking, as used in manipulating robots [19], [22] or manipulating a mouse pointer by moving the head [6], [14]. Recognition methods also provide quite a large number of tools for analysing this kind of objects. However, none of the aforementioned methods is based on the direction field analysis, which in a easy way can locate the central point of the cylindrical object and then approximate its contour.

## 2 Aim of the Work

The goal of our work is to propose an algorithm for simplified analysis of an iris using the direction field [9]. Then, this approach is generalized for the detection of outlines of any cylindrical objects. Eventually, we verify the qualities and accuracy of the assumed form of the algorithm by applying it for the segmentation of an iris.

## 3 The Directional Field

There are many methods for calculating the directional field. One of them consists in calculating two matrices from a grayscale image. The first of the matrices indicates the fit level of the searched object, the second one - the angle under which the best fit was found. This kind of analysis employed for elongated objects was presented in detail in [9], [10]. Below we present its fundamental properties, which, in further chapters, are extended to cylindrical objects; we also propose a method of the segmentation of such objects. Assume an grayscale input image  $L_{GRAY}$  of the resolution  $M \times N$  (where  $M$  - number of rows,  $N$  - number of columns) that contains at least one cylindrical object. By a cylindrical object we understand an object that contains at least one focus formed by any set of pixels that the focus encloses (in general, the focus need not to be located in the image). Additionally, the object can have a contour, either closed or opened.



**Fig. 1.** The values of the mask  $h$  for the  $\lambda = 0^\circ$  angle

degree  $\lambda$ ), for angles  $\lambda$  from the range of  $0^\circ$  to  $360^\circ$ , with the  $step_T$  (for example, for our purposes, every  $10^\circ$ )

$$L_{GRAY_h}(m, n, \lambda) = \sum_{m_h = -M_h/2}^{M_h/2} \sum_{n_h = -N_h/2}^{N_h/2} L_{GRAY}(m + m_h, n + n_h) \cdot h_\lambda(m_h, n_h) \quad (1)$$

This object will be filtered with a median filter with the mask size adjusted to the size of the object (i.e. its resolution), and then an iterative process will calculate a convolution for angles  $\lambda$ , from the range of  $0^\circ$  to  $360^\circ$ , using a convolution mask  $h$  of resolution  $M_h \times N_h = 20 \times 10$ . An example of mask values for  $\lambda = 0^\circ$  is presented below (fig. 1).

The next step of the algorithm is performing sequential convolution with mask  $h_\lambda$  (obtained after rotating  $h$  by the

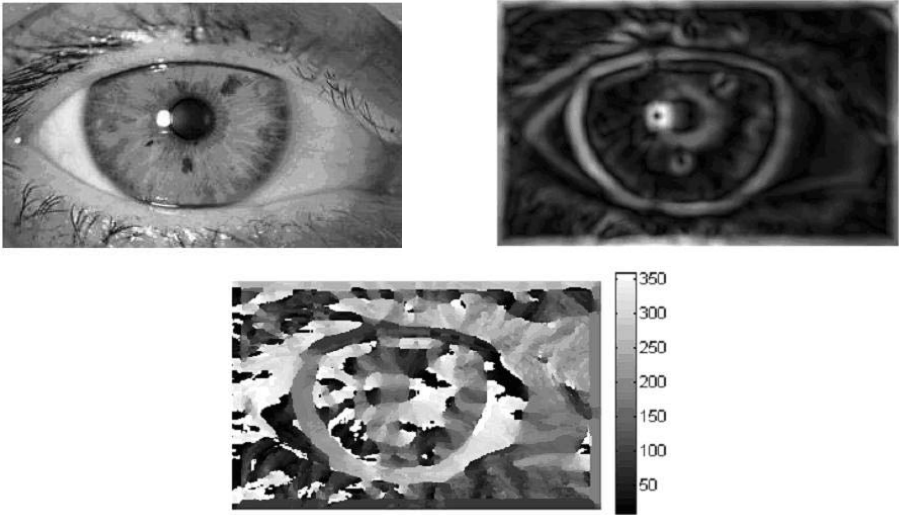
Next, the maximum value for every pixel of the created images  $L_{GRAY_h}$  is calculated. During this operation the information for which  $h$  (i.e. for which  $\lambda$  angle) the maximum value was obtained is recorded.

$$L_{ma}(m, n) = \max_{\lambda \in (0, 360]} (L_{GRAY_h}(m, n, \lambda)) \quad (2)$$

The image  $L_\lambda$  is an image of resolution the same as  $L_{ma}$ , with the pixel values representing the degrees of  $\lambda$  for which the maximum was reached (formula (2)), i.e.:

$$L_\lambda(m, n) = \begin{cases} \lambda & \text{for } L_{GRAY_h}(m, n, \lambda) = L_{ma}(m, n) \\ L_\lambda(m, n) & \text{for } L_{GRAY_h}(m, n, \lambda) \neq L_{ma}(m, n) \end{cases} \quad (3)$$

for  $\lambda \in (0, 360]$  The obtained images  $L_\lambda$ ,  $L_{ma}$  and the intermediate phases of  $L_{GRAY_h}$  are shown in fig. 2.



**Fig. 2.** The consecutive steps of the algorithm: a) the input image -  $L_{GRAY}$ , b) the  $L_{ma}$  image, c) the  $L_\lambda$  image)

The degree values obtained in  $L_\lambda$  are the key element for further calculations. As it is shown in fig. 2 c), these values, according to possible angles of rotation of mask  $h_\lambda$ , belong to the set  $0, 10, 20, \dots, 350, 360^0$ . In practice, increasing the resolution of the measurement of the angle means increasing the number of iterations, that is decreasing the step value for  $\lambda$ . The area in which the search and pixel verification is performed is narrowed by assuming a binarization threshold normalized to the full range of gray levels of the  $L_{ma}$  image and then multiplying the obtained image with the  $L_\lambda$  image (fig. 3):

$$L_{\lambda b}(m, n) = L_\lambda(m, n) \cdot L_{BIN}(m, n) \quad (4)$$



Fig. 3. Image  $L_{\lambda b}$

the line perpendicular to and associated with the given pixel, and the lines in other points, according to fig. 4 a).

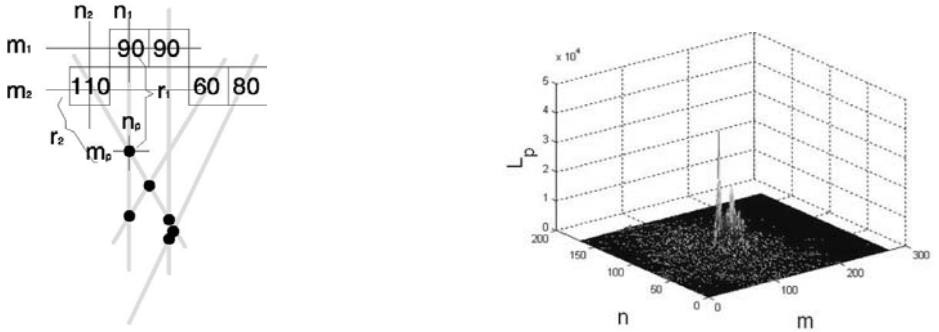


Fig. 4. A diagram of the modified method of calculating the points  $m_p, n_p$  and an example of an accumulation image  $L_p$

Therefore if the coordinates of the analyzed pair of pixels are recorded as  $(m_1, n_1)$  and  $(m_2, n_2)$ , then we use the equation of the straight line:

$$b_1 = m_1 - k_1 \cdot n_1 \quad (5)$$

$$b_2 = m_2 - k_2 \cdot n_2 \quad (6)$$

where

$k_1$  and  $k_2$  - equal, respectively,  $tg(\lambda_1)$  and  $tg(\lambda_2)$ ,

$m_1, m_2, n_1, n_2$  - coordinates of row and column of the analyzed pixels.

The point  $(m_p, n_p)$  of intersection between the lines is obtained from the equation:

$$m_p = \frac{b_1 \cdot k_2 - b_2 \cdot k_1}{k_2 - k_1} = \frac{m_1 \cdot k_2 - n_1 \cdot k_1 \cdot k_2 - m_2 \cdot k_1 + n_2 \cdot k_1 \cdot k_2}{k_2 - k_1} \quad (7)$$

$$n_p = \frac{b_2 - b_1}{k_1 - k_2} = \frac{m_2 - k_2 \cdot n_2 - m_1 + k_1 \cdot n_1}{k_1 - k_2} \quad (8)$$

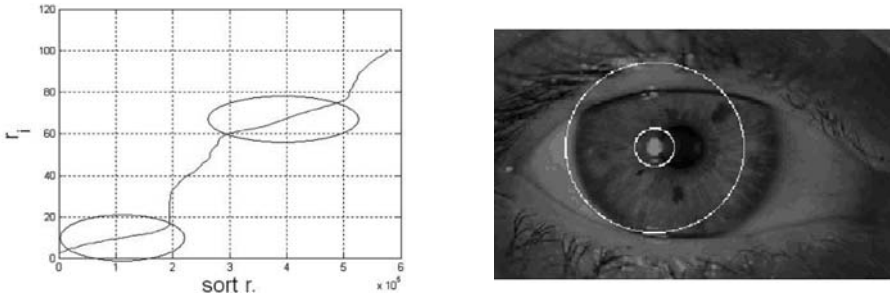
On this basis, for consecutive pixel values an accumulation image  $L_p$  is created, which contains (at first, for the first iteration in all pixels "0") information about the location of focuses for consecutive iterations  $j$ , in the following way:

$$L_p(m, n)^{(j+1)} = L_p(m, n)^{(j)} + 1 \text{ for } m = m_p, n = n_p \quad (9)$$

Fig. 4 b) shows results obtained for an exemplary matrix  $L_{\lambda b}$ . The next step of the algorithm is finding the radius of a selected focus. fig. 4 b) shows two maximum values of matrix  $L_p$ . Considering one of them, we can show sorted values of radii  $r_i$

$$r_i = \sqrt{(m_1 - m_p)^2 + (n_1 - n_p)^2} \quad (10)$$

for lines intersecting at this point - fig. 5.



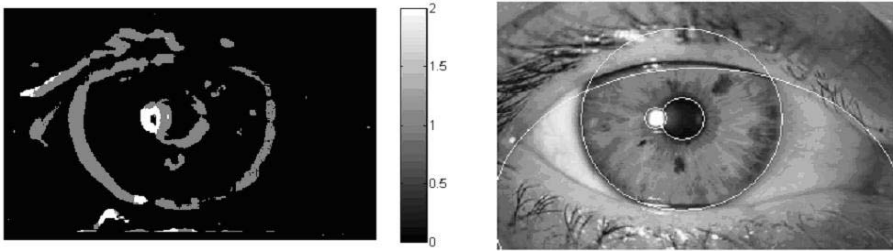
**Fig. 5.** Graph of changes in values of radii  $r_i$  for intersecting lines. The ellipses mark pseudostable radius values. b) circles on the input image, created on the basis of radius values  $r_{s1} = 15$  and  $r_{s2} = 68$ .

The graph obtained does not confirm sufficient 'stability' of radius value  $r_i$  for any of its values. The only visible ranges where the radius value seems acceptable are  $10 \div 20$  and  $60 \div 75$  pixels (marked with ellipses on fig. 5). Results obtained for average values in marked ranges, i.e.  $r_{s1} = 15$  and  $r_{s2} = 68$  are acceptable as shown on fig. 5 b). Therefore, on the basis of images  $L_p$  and  $L_{\lambda b}$  we can try to reconstruct the contour.

### 3.1 Detecting the Contour of a Cylindrical Object

The literature [8], [14], [21] at many times describes a possibility of detecting (reconstruction) of the contour of cylindrical objects with strong noise. None of the described methods was performed for the inclination field, except [2] i [13] which describe a solution different to that presented below. Images  $L_p$  and  $L_{\lambda b}$  containing information about points of intersection between lines to all pixels, and information ( $L_p$ ) about the number of these intersections in a given point, determine (as shown on fig. 5) the positions of focuses of cylindrical shapes. As can be seen (fig. 5), these positions are identified definitely, but determining





**Fig. 6.** a) Image after segmentation  $L_{CON}$  and a palette of grey values corresponding to labels 0, 1, 2 b) the obtained contour on input image  $L_{GRAY}$

the radius value is not a definitely defined problem. One of the methods for improving the stable value of radius  $r_i$  is using the information about the content of image  $L_{GRAY}$  and creating a matrix  $L_{GLCM}$  (Grey Level Co-occurrence Matrix) for areas of the size  $10 \times 10$  and then a contrast matrix  $L_{CON}$  (sum of squares variance), for segmentation of areas of image  $L_{\lambda b}$ . Fig. 6 shows labels given to specific areas  $L_{\lambda b}$  after segmentation. For each of visible areas (fig. 6 a - specific labels) the value of radius  $r_{si}$  is calculated separately and the contour is automatically detected. Results are shown on fig. 6 b where the contour is marked with ranges. As the presented results of the algorithm show, contours of all larger objects have been found. Any possible errors come from the deviations of actual objects from the shape of circle or ellipse.

## 4 Conclusion

The presented application of the method of detecting the contour of cylindrical objects described here is ophthalmology in medicine. The areas of interest here are the inside of the iris and the white of an eye. The paper additionally presents and verifies a method of approximating the contour of cylindrical objects with circles, with the use of direction field. At present the algorithm is being extended into objects of different type where the dominant element is the segmentation and reconstruction of edges of objects with strong noise. Efforts are also taken to decrease the number of the steps of the algorithm presented.

## References

1. Ahlberg J.: A system for face localization and facial feature extraction, Technical Report (No. LiTH-ISY-R-2172), Linkoping University, Sweden, 2002.
2. Ajdari A. R., Faez K., Qaragozlou N.: Fast Circle Detection Using Gradient Pair Vectors, VIIth Digital Image Computing: Techniques and Applications, Sydney, 2003.
3. Donoho D., Huo X.: Beamlet pyramids: A new form of multiresolution analysis, suited for extracting lines, curves, and objects from very noisy image data. In Proceedings of SPIE, volume 4119, July 2000.

4. Donoho D., Huo X.: Beamlets and multiscale image analysis. In: T.J. Barth, T. Chan, and R. Haimes (Eds.), Springer Lecture Notes in Computational Science and Engineering-ing 2001, 149-196.
5. Duda R.O., Hart P.E.: Use of the Hough Transformation to detect lines and curves in pictures, *Communications of the Association of Computing Machinery* 15 (1972) 11-15.
6. Gorodnichy D., Malik S., Roth G.: Affordable 3D face tracking using projective vision. In Proc. Intern. Conf. On Vision Interface (VI'2002), Calgary, May 2002.
7. Hough P. V. C.: Method and means for recognizing complex patterns, U.S. Patent 3069654, 1962
8. Kass M., Witkin A., Terzopoulos D.: Snake: active contour models, *Int. J. Computer Vision*, 1 (4), 1987, 321-331
9. Koprowski R., Wrobel Z.: Analysis of the inclination of elongated biological objects - microtubules, sent to *Medical Image Analysis*, 2006.
10. Koprowski R., Wrobel Z.: Estimation of the properties of algorithms for measuring the inclination angle of elongated objects, sent to *Image and Vision Computing*, 2006.
11. Korepanov A. O. Central lines extraction on the diagnostic vessel images using the methods of wavelet-analysis and differential geometry, 7 th, International Conference on Pattern Recognition and Image Analysis: New Information Technologies, PRIA-7-2004, Volume III, 740-743.
12. Lacroix V., Acheroy M.: Feature-extraction using the constrained gradient. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(2):April 1998.
13. Libor M.: Recognition of Human Iris Patterns for Biometric Identification, PhD, Institute of Automation, Chinese Academy of Sciences, 2003.
14. Lien J.J., Kanade T., Cohn J., Li C.: Detection, tracking, and classification of subtle changes in facial expression, *Journal of Robotics and Autonomous Systems*, Vol. 31, 131 - 146, 2000.
15. Mian Z., Kassim A. A., Mannan M. A., Texture Analysis Of Machined Surfaces Using A New Hough Transform, *Proceedings of Texture 2002 - The 2nd international workshop on texture analysis and synthesis*, pp. 35-41, June 1st, 2002, Copenhagen
16. Rosenfeld S. Thurston M., Edge and curve detection for visual scene analysis, *IEEE Trans. Comput. C-20*, pp. 562-569, 1971.
17. Serra. J.: *Image Analysis and Mathematical Morphology*. Volume 2: Theoretical Advances. Academic Press, 1988.
18. Song J., Cai M., Lyu M. R., Cai S.: A New Approach for Line Recognition in Large-size Images Using Hough Transform, *International Conference on Pattern Recognition 2005*.
19. Tian Y.L., Kanade T., Cohn J.F.: Recognizing action units for facial expression analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 97-116, 2001
20. Ting-jen Y.: A Qualitative Profile-based Approach to Edge Detection (Near-Optimal Detection of Geometric Objects by Fast Multiscale Methods), Phd, Department of Computer Science New York University, September 2003.
21. Xu C., Prince J. L.: Gradient vector flow: a new external force for Snake, *IEEE Proc. Conf. on CVPR'97*, 1997, 66-71
22. Yang M., Ahuja N., Kriegman D.: Detecting faces in mages: A survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(1): 2002, 34-58.

23. Yuanxin Zhu, Bridget Carragher, David Kriegman, and Clinton S. Potter Perceptual Organization as a Method for Detection and Selection of Filamentous Structures in Highly Noisy Images Acquired by Cryoelectron Microscopy *Journal of Structural Biology*, 2001.
24. Yuille A., Hallinan P., Cohen D.: Feature Extraction from Faces Using Deformable Templates, *International Journal of Computer Vision*, 8(2): 99-111, 1992.

---

# Trajectory Fusion for Multiple Camera Tracking

Ariel Amato<sup>1</sup>, Murad Al Haj<sup>1</sup>, Mikhail Mozerov<sup>1</sup>, and Jordi González<sup>2</sup>

<sup>1</sup> Computer Vision Center and Department d'Informàtica. Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain  
aamato@cvc.uab.es

<sup>2</sup> Institut de Robòtica i Informàtica Industrial(UPC-CSIC), Edifici U Parc Tecnològic de Barcelona. 08028, Spain

**Summary.** In this paper we present a robust and efficient method to overcome the negative effects of occlusion in the tracking process of multiple agents. The proposed approach is based on the matching of multiple trajectories from multiple views using spatial and temporal information. These trajectories are represented as consecutive points of a joint ground plane in the world coordinate system that belong to the same tracked agent. We introduce an integral distance between compared trajectories, which allows us to avoid mismatches, due to the possible measurement outliers in one frame. The proposed method can also be considered as an interpolation algorithm of a disconnected trajectory during the time of occlusion. This technique solves one of the most difficult problems of occlusion handling, which is a matching of two unconnected parts of the same trajectory.

## 1 Introduction

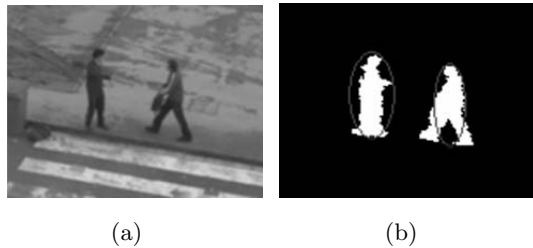
Tracking objects with multiple views is an active area of research that finds application in automated surveillance, video archival-retrieval, and humancomputer interaction. This work presents a method based on the matching of multiple trajectories of the same agent which are obtained from multiple views. This approach allows to resolve the temporal discontinuity of a trajectory, which occur due to the occlusion. Consequently, trajectories obtained from different cameras are fused in the same ground plane. Occlusion handling is a complex task that has been thoroughly investigated [1], which can be classified in four areas: region based tracking [2], active contour-based tracking [3], feature-based tracking [4], and model-based tracking[5]. Existing approaches include Extended Kalman Filters (EKF) was used for occlusion analysis [6]. A relationship between camera views using homography is considered in [7]. To resolve mutual occlusion the best view is utilized in [8]. A similar method that based on a fusion of information from every camera into a 3-D geometric coordinate system is presented in the work [9]. Most of such techniques need high computationally cost, and the accuracy depends on the different object conditions. In contrast, our approach needs just a few ground plane points of tracked agents in each frame. Another advantage of the presented method is the strong mathematical conclusion of

the tracking process, which easily allows implementing the proposed model to the arbitrary scene. Our experiments with a real data show that it is possible to track the trajectories very effectively even in the presence of occlusion using only the ground plane points of the agents.

This paper is organized as follows. Section 2 describes the method used for segmentation objects. Section 3 describes how to obtain the overlapping camera view process in a joint ground plane, separated in two step, camera calibration and top view conversion. Section 4 shows the main process in order to obtain time continue trajectory. In section 5, experimental results are shown.

## 2 System Input Data

Our tracking process involves on stage of segmentation. We choose the method of colour background subtraction that is described in [10]. Once the segmentation is performed, the size and position of the ellipsoid that surrounds the object is obtained. The floor point projection can be calculated using intrinsic and extrinsic parameter of each camera. The result of segmentation is shown in Fig. 1.



**Fig. 1.** (a) Original Image (b) Segmentation with foot point

## 3 Data Fusion

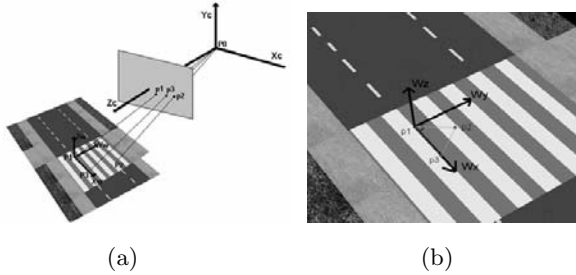
To convert the image planes obtained from multiple camera views into one ground plan map of an inspected scene, a calibration process is needed.

### 3.1 Camera Calibration

Most traditional camera calibration techniques require specific knowledge about the geometric characteristics of the referenced object such as Direct Linear Transformation (DLT) [11] which solves the perspective matrix linearly; similar methods include Tsais [12] and Zhangs [13]. Plane based methods [14] use the same DLT paradigm, but show more flexibility. The method we use in this paper is based on geometric derivation [15]. We choose this because it uses only



**Fig. 2.** Two viewpoints of the same scene



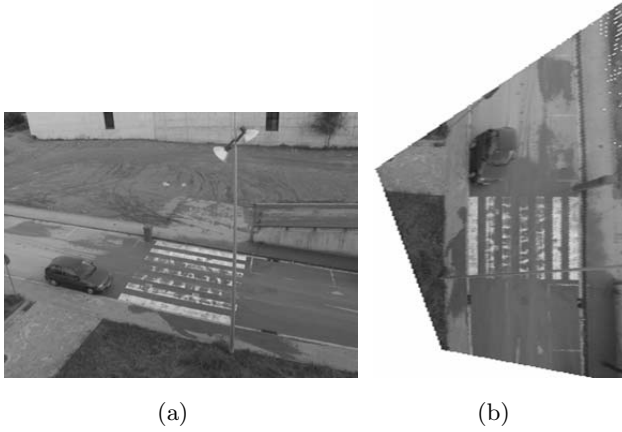
**Fig. 3.** (a) Camera model and ground plane (b) New coordinate systems

distance between three points in the world coordinate system to achieve calibration. Rotation matrix  $\mathbf{R}$  and translation vector  $\mathbf{T}$  for each camera coordinate system are obtained via the given distance between the vertices of the marker triangle formed by the three points of reference.

The intrinsic parameter like focal radio can be obtained independently; we could assume that the focal radio is known (from the data sheet) or calculated in a simple experimental process with two known points and the distances between these points and the camera. Two different points of views of the same scene with the landmark of three points are represented in Fig. 2. The pin-hole camera model is shown in Fig. 3(a).

### 3.2 Joint Ground Plane

To match the points from different points of view, a joint coordinate system is needed. Once the segmentation algorithm localizes a pixel of interest  $(i,j)$  in the image of one camera (in our case it is a foot point of an agent), this pixel should be projected onto the joint plane in the world coordinate system for matching with another point in a different image obtained from a second camera. This projection process is referred to as the Inverse Perspective Mapping (IPM). Fig. 4 show an example of the IPM in a complete image.



**Fig. 4.** Top View Transformation (IPM)

## 4 Trajectory Tracking

Initially, we have a set of  $A$  cameras  $C_\alpha$ , and  $T$  images  $I_{(t,\alpha)}$  obtained from each camera. The segmentation algorithm plus the joint ground plane transformation provide us with a set of floor points:

$$\mathbf{Q} = \bigcup_{A,T,N(\alpha,t)} \mathbf{p}_{\alpha,n}^t \quad (1)$$

where  $p$  is a point of the world coordinate system in the joint ground plane. The number of points in a single image at a certain snapshot time  $t$  depends on the camera index  $\alpha$ , due to occlusion and non overlapped fields of view of different cameras. So, it is convenient to complement the set of real points  $\mathbf{Q}$  by a set of invisible or imaginary points

$$\tilde{\mathbf{Q}} = \bigcup_{A,T,\tilde{N}(\alpha,t)} \tilde{\mathbf{p}}_{\alpha,n}^t \quad (2)$$

where the coordinates of point  $\tilde{p}$  are calculated using some rules that will be explained below. An example of such extension is illustrated in Fig. 5. It is useful to define a set of points that belong to the same time of the cameras snapshot as

$$\mathbf{P}_{m(t)}^t = \bigcup_A \mathbf{p}_{\alpha,n(m)}^t \quad (3)$$

where;  $1 \leq m(t) \leq \max_{\alpha} \{N(\alpha, t)\}$  and the number of points in this set always is equal to number of cameras, which means that the set can include also imaginary points.

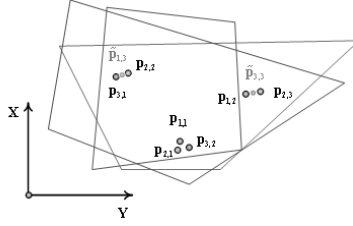


Fig. 5. Overlapping View in a joint plane

Our main goal now is to associate all these points with several tracks.

$$\mathbf{O}_k = \bigcup_T \mathbf{P}_k^t \quad (4)$$

Where  $1 \leq k \leq K$ , and, in principle,  $\max_{\alpha,t} \{N(\alpha,t)\} \leq K$ , which means that a certain track can include void elements in Eq.( 3), in other words, the time domain of a track possibly cannot coincide with the full interval of the analyzed scene performance  $[1,T]$ . We introduce distance  $\Delta_k^t$  between two consecutive sets in the same track as

$$\Delta_k^t = \|\mathbf{P}_k^t - \mathbf{P}_k^{t-1}\| = \sum_{\alpha \in A} \left\| \mathbf{p}_{\alpha,n(k)}^t - \mathbf{p}_{\alpha,n(k)}^{t-1} \right\| \quad (5)$$

And the intrinsic distance of a set  $\mathbf{P}_k^t$  itself:

$$D_k^t = \sum_{i < j < A} \left\| \mathbf{p}_{i,n(k)}^t - \mathbf{p}_{j,n(k)}^t \right\| \quad (6)$$

It is reasonable to admit that for the track of a real agent these distances have to be minimal. So, now we formulate an optimization problem as follows: find the sequences of points of a track, which minimize the sum of distances:

$$\mathbf{O}_k = \arg \min_{\mathbf{P}_k} \sum_{t \in T} (D_k^t + \Delta_k^t) \quad (7)$$

This problem can be solved recurrence growing of each track:

$$\mathbf{O}_k^\tau = \mathbf{O}_k^{\tau-1} \bigcup \mathbf{P}_k^\tau \quad (8)$$

where

$$\mathbf{O}_k^\tau = \bigcup_{t=1}^{\tau} \mathbf{P}_k^t \quad (9)$$

$$\mathbf{P}_k^\tau = \arg \min_{\mathbf{p}_{\alpha,n}^\tau} \{D_k^\tau + \Delta_k^\tau\} \quad (10)$$

The expression of Eq.( 10) can be easily solved by direct search of the minimum of the objective function  $D_k^\tau + \Delta_k^\tau$ ;

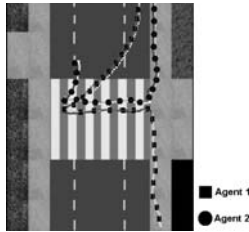


## 5 Experimental Results

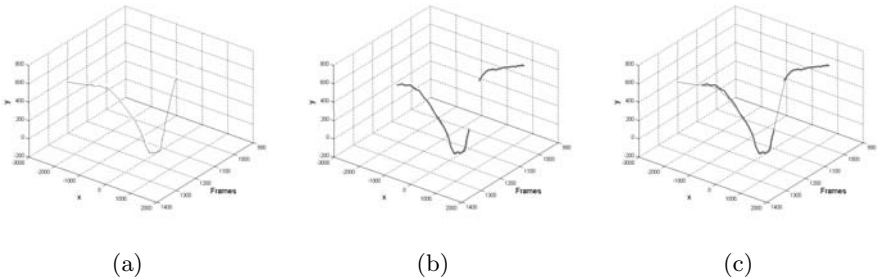
Our experiments were conducted with two cameras. To illustrate the result of tracking we chose a situation with only two agents. Camera 1 was able to track both agents without occlusion, but camera 2 was unable to do so because one agent occluded the second for a long time. In Fig. 6(a),(b) show two snapshots from camera 1 and 2 respectively. Fig. 6(c) shows the trajectory of both agents, captured from camera 1, in joint ground plane. In Fig. 6(d) the same is shown but from camera 2.



**Fig. 6.** (a) Snapshot camera 1 (b) SnapShot camera 2 (c) Trajectory in joint ground plane camera 1 (d) Trajectory in joint ground plane camera 2



**Fig. 7.** Overlapping Trajectories camera1 camera 2



**Fig. 8.** (a) Trajectory of agent 1 in (time -space) domain of camera 1 (b) Trajectory of agent 1 in (time -space) domain of camera 2 (c) Trajectory overlapping Camera 1 and Camera 2 in (time-space) domain

When we applied image matching, the discontinuity is resolved and then our algorithm can track efficiently the trajectory Fig. 7.

We tested our method with more than 5000 frames with occlusion and multiple agents, and we were able to track correctly all the trajectories. Fig. 8 (a) and (b) shows one occluded agent trajectory in time-space domain, from camera 1 and camera 2 and Fig. 8 (c) shows the result of trajectory interpolation.

## 6 Conclusion

This paper presents a method for resolving occlusion of tracked agents using multiple views. The experimental results show that it is possible to track the trajectories effectively, even in the presence of occlusion, using only the ground plane points of the agents. This method uses strict mathematical model, which makes the tracking process robust and highly efficient.

## Acknowledgements

This work is supported by EC grants IST-027110 for the HERMES project and IST-045547 for the VIDi-video project, and by the Spanish MEC under projects TIN2006-14606 and DPI-2004-5414. Jordi Gonzàlez also acknowledges the support of a Juan de la Cierva Postdoctoral fellowship from the Spanish MEC.

## References

1. YWeiming Hu, Tieniu Tan, Fellow, IEEE, Liang Wang, and Steve Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors", *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions*, vol 34, pp. 334–352, 2004.
2. M. Kilger, "A shadow handler in a video-based real-time traffic monitoring system", *IEEE Workshop Applications of Computer Vision Palm Springs, CA*, pp. 11–18, 1992.
3. L. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components", *IEEE Trans. Pattern Recognit. Machine Intell.*, vol. 23, pp. 349–361, 2001.
4. R. Polana and R. Nelson, "Low level recognition of human motion", *IEEE Workshop Motion of Non-Rigid and Articulated Objects*, pp. 77–82, 1994.
5. Dae-Sik Jang, Hyung-Il Choi, "Active models for tracking moving objects", *Pattern Recognition*, vol. 33, no. 7, pp.1135–1146, 2000.
6. Rosales, Sclaroff, "Improved Tracking of Multiple Humans with Trajectory Prediction and Occlusion Modeling", *IEEE Conference on CVPR, Workshop on the Interpretation of Visual Motion, Santa Barbara, CA*, pp. 117–123, 1998.
7. A. Stein, "Tracking from Multiple View Points: Selfcalibration of Space and Time", *In DARPA IU Workshop*, pp. 1037–1042, 1998.
8. A. Utsumi, H. Mori, J. Ohya, and M. Yachida., "Multiple-view-based tracking of multiple humans", *in Proc. Int. Conf. Pattern Recognition*, pp. 197–601, 1998.

9. R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade., “Algorithms for cooperative multi-sensor surveillance”, *Proceedings of the IEEE*, vol. 89 pp. 1456–1477, 2001.
10. I.Huerta and D. Rowe and M. Mozerov and J. Gonzàlez, “Improving Background Subtraction based on a Casuistry of Colour-Motion Segmentation Problems”, *3rd IbPRIA, Gerona, Spain*, vol. 2 pp. 475–482, 2007.
11. Y. I. Abdel-Aziz, H.M. Karara, “Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry”, *Proc., of the ASP Symposium on Close-Range Potogrammetry*, pp.1–18, 1971.
12. R. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using offthe-shelf TV cameras and lenses”, *IEEE J. Robot. Autom.*, vol 3 pp. 323–344, 1987.
13. Z. Zhang, “A flexible new technique for camera calibration”, *LIEEE Trans. Pattern Analysis and Machine Intelligence*, vol 22 pp.1330–1334, 2000.
14. L. Lucchese, “Geometric calibration of digital cameras through multiview rectification”, *Image Vis. Comput*, pp. 517–539, 2005
15. Huang T., Netravali, A., “Motion and Structure from Feature Correspondences”, *Proceedings of the IEEE*, vol 82 pp. 252–268, 1994.

---

# Object Selection Based on Clustering and Border Objects

J. Arturo Olvera-López, J. Ariel Carrasco-Ochoa,  
and J. Francisco Martínez-Trinidad

National Institute of Astrophysics, Optics and Electronics,  
Luis Enrique Erro No. 1, Sta. María Tonantzintla, Puebla, CP: 72840, Mexico  
{aolvera, ariel, fmartine}@ccc.inaoep.mx

**Summary.** Object selection is an important task for instance-based classifiers since through this process the time in training and classification stages could be reduced. In this work, we propose a new method based on clustering which tries to find border objects that contribute with useful information allowing to the classifier discriminating between classes. An experimental comparison of our method, the CLU method based on clustering, and the DROP methods, is presented.

## 1 Introduction

In Pattern Recognition, the supervised classification process assigns a class or label to new objects according to their features using a set of previously assessed objects called training set. In this work, we denote the training set as  $T$ .

In practice, some information in  $T$  could be non relevant since  $T$  commonly contains superfluous objects, that is, objects with useless information for the classification task.

For instance-based classifiers (classifiers that use the whole set  $T$  to classify each new object), training and classification are expensive tasks. Therefore it is necessary to select those objects (in  $T$ ) that give relevant information for the classifier. This selection process is known as object selection which obtains a set  $S \subset T$  such that  $S$  contains relevant objects that are chosen through a selection criterion.

In this paper, we propose a method for object selection based on clustering, our method finds and retains border objects. In a training set, the objects located at the interior of a class (central objects) are superfluous since their deletion does not affect the classification accuracy. On the other hand, the objects which lie on borders between classes (border objects) give useful information for the classifier to preserve the class discrimination regions [1, 2]. In order to show the performance of the proposed method, we present an experimental comparison between our method and some other object selection methods analyzing their behavior using the objects set obtained as training for different classifiers.

The paper is organized as follows: in 2, we describe some methods for solving the object selection problem. In 3, our object selection method is proposed, in 4, we report experimental results obtained using our method, and finally, in 5, we present some conclusions and directions for future work.

## 2 Related Work

In this section, we describe some relevant methods for solving the object selection problem.

The Condensed Nearest Neighbor (CNN) [3] and the Edited Nearest Neighbor (ENN) [4] are two of the first proposed methods for object selection. The CNN method starts with  $S=\emptyset$  and its initial step consists in randomly including in  $S$  one object belonging to each class. After the initial step, each object in  $T$  is classified (with  $k$ -Nearest neighbor,  $k$ -NN) using  $S$  as training set, if an object  $O$  is misclassified then  $O$  is included in  $S$  to ensure that new objects near to  $O$  will be classified correctly.

The ENN rule consists in discarding from  $T$  those objects that do not belong to their  $k$  nearest neighbors' class. This method is used as noise filter because it deletes noisy objects, that is, objects with a different class in a neighborhood. An extension of ENN is REEN (Repeated ENN) [5] which applies ENN repeatedly until all objects in  $S$  have the same class that the majority of their  $k$  nearest neighbors.

The DROP (Decremental Reduction Optimization Procedure) methods were proposed in [1], their selection criterion is based on the concept of associate. The associates of an object  $O$  are those objects such that  $O$  is one of their  $k$  nearest neighbors. DROP1 starts with  $S=T$  and discards  $O$  if its associates in  $S$  can be classified correctly without  $O$ . DROP2 considers the effect of the removal of an object on all the training set  $T$ , DROP2 discards  $O$  if its associates in  $T$  can be classified correctly without  $O$ . DROP3 and DROP4 apply a noise filter (similar to ENN) before starting the selection process. DROP5 modifies DROP2 so that the selection process starts with the nearest enemies (nearest objects with different class).

In [2] the Iterative Case Filtering algorithm (ICF) was proposed, this method is based on the  $\text{Reachable}(O)$  and  $\text{Coverage}(O)$  sets which are the neighborhood set and the associates set described above. ICF discards an object  $O$  if  $|\text{Reachable}(O)| > |\text{Coverage}(O)|$ .

Clustering is a non supervised technique which has been used for object selection. In [6] the CLU object selection method which is based on clustering was proposed and applied to the signature recognition problem. CLU divides the training set in  $n$  clusters and then  $S$  is the set of centroids of each cluster.

The methods described above (excluding CLU) and most of the object selection methods have been proposed following the nearest neighbor rule. Therefore, the object sets obtained by these methods have a good performance using the  $k$ -NN classifier, but these sets could be non suitable for other classifiers.

### 3 Proposed Method

As we mentioned in section 1, central objects could be deleted from  $T$  without loosing classification accuracy. In this paper we propose a method called OSC (Object Selection by Clustering) which finds and retains border objects and some central objects. The selection criterion in OSC is based on clustering, mainly on non homogeneous clusters.

An homogeneous cluster is an object set such that all objects belong to the same class whereas in a non homogeneous cluster there are objects belonging to different classes.

In order to find border objects, the OSC method generates clusters and analyzes non homogeneous clusters since border objects are located in critical regions, that is, those regions which contain closer objects belonging to different classes. The OSC method is shown in figure 1.

---

```

OSC (Training set  $T$ , number of clusters  $n$ ): object set  $S$ 
 $S = \emptyset$ 
 $Clust = Gen\_clust(T, k)$  //create  $n$  clusters from  $T$  //
For each cluster  $A_j$  in  $Clust$ 
  If  $A_j$  is homogeneous then
     $m = \text{mean of cluster } A_j$ 
    Find the object  $O_i$  nearest to  $m$ 
     $S = S \cup O_i$ 
  Else //  $A_j$  is non homogeneous //
    Find the majority class  $C_M$  in cluster  $A_j$ 
    For each class  $C_k$  in  $A_j$  ( $C_k \neq C_M$ )
      For each object  $O_j$  belonging to class  $C_k$ 
        Find  $O_c \in C_M$ , the nearest object to  $O_j$  with class  $C_M$ 
         $S = S \cup \{O_c\}$ 
        Find  $O_M$ , the nearest object to  $O_c$  with class different to  $C_M$ 
         $S = S \cup \{O_M\}$ 
Return  $S$ 

```

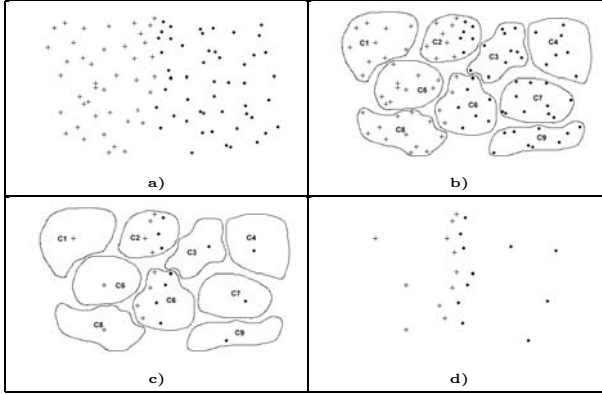
---

**Fig. 1.** OSC method for object selection

OSC starts creating  $n$  clusters (function  $Gen\_clust$ , which can be any clustering method) from  $T$ . After that, for each cluster  $A_j$  it is necessary to decide whether  $A_j$  is homogeneous or not.

If  $A_j$  is homogeneous then the objects in  $A_j$  are central objects, that is, they do not lie on critical regions therefore they could be discarded from  $T$  without affecting the classification accuracy. OSC finds the nearest object to the mean (mean object) and discards the remaining objects from  $A_j$  so that  $A_j$  is represented by the mean object.

If  $A_j$  is non homogeneous then there are in  $A_j$  some objects located at critical regions, that is, border objects. In order to find the border regions, OSC finds the minority class objects (objects that do not belong to the most frequent class in  $A_j$ ) since these objects are nearest to a border delimited by the most frequent class in  $A_j$ . Once the minority class objects have been found, the border objects in the most frequent class are the nearest objects to each minority class object, and by analogy, the border objects in the minority class are the nearest objects to each border object in the majority class.



**Fig. 2.** a) Dataset with classes "+" and "•". b) Clusters created from the dataset. c) Objects selected in each cluster. d) Objects set obtained by OSC.

Finally, the object set selected by OSC consists of the mean objects and the border objects described above.

To illustrate how OSC finds border objects let us consider the sample shown in figure 2a which is a bi-dimensional dataset with objects belonging to the classes "+" and "•". The clusters (C1...C9) created from the dataset are depicted in figure 2b in which the non homogeneous clusters are C2 and C6.

In cluster C2, the minority class is "•", then the border objects in the most frequent class ("+") are the nearest objects to each minority class object (•). On the other hand, the border object in class "•" are the nearest objects (belonging to class "•") to each border object in class "+".

In the cluster C6, the minority class is "+" and OSC follows the same process described before. In figure 2c, the objects selected in each cluster are depicted and figure 2d shows the objects set obtained by OSC. We can observe that OSC finds border objects and some central objects (mean objects in the homogeneous clusters).

## 4 Experimental Results

In this section, we report the results obtained applying our proposed OSC method over ten datasets from the UCI dataset repository [7]. In addition, we show a performance comparison among OSC, CLU and the DROP methods because according to the results reported in [1] and [2], the DROP methods outperform to other relevant object selection methods such as ENN, RENN and ICF. For all the experiments, 10 fold cross validation was used.

For OSC and CLU methods it is necessary to generate  $n$  clusters,  $n > c$ , where  $c$  is the total number of classes in the dataset. In order to choose the value of  $n$  to be used in our experiments we carried out an experiment over ten datasets using different  $n$  values to choose the best ones where OSC and CLU (respectively)

have the best performance in the average case. In table 1 we show the classification accuracy obtained by OSC and CLU using the values  $n=c$ ,  $4c$ ,  $6c$ ,  $8c$  and  $10c$ . For creating clusters, the k-means algorithm was used. For testing the object sets selected by OSC and CLU, the  $k$ -NN classifier ( $k=1$  using Euclidean distance) was used. Also we show the classification obtained by the original training set (*Orig.*). Based on the results shown in table 1 we can observe that the best value for  $n$  using OSC was  $n=6c$  and the best one for CLU was  $n=8c$ , so these values were used in all the next experiments reported in this section.

**Table 1.** Classification obtained by *CLU* and *OSC* using different number of clusters

Dataset	Number of clusters										
	Orig.	$n=c$		$n=4c$		$n=6c$		$n=8c$		$n=10c$	
		CLU	OSC	CLU	OSC	CLU	OSC	CLU	OSC	CLU	OSC
Glass	73.31	48.63	62.53	59.74	68.61	59.80	69.50	63.48	69.13	62.89	67.14
Iris	95.33	80.00	89.33	90.00	92.00	93.33	93.33	90.00	92.66	85.70	94.00
Liver	61.99	46.94	57.74	50.15	55.34	54.24	57.63	58.24	55.33	59.63	56.78
Pendigitis	99.42	71.11	94.30	90.23	95.85	92.84	96.78	95.58	96.93	95.86	97.10
Segmentation	95.85	89.85	81.23	70.66	88.23	74.28	89.85	84.52	92.71	79.34	90.57
Sonar	72.64	50.02	64.92	48.45	63.92	49.90	65.85	56.73	63.00	56.78	62.00
Spambase	82.41	47.17	68.81	68.31	69.61	67.07	69.31	66.13	69.68	58.11	69.91
Thyroid	95.37	79.54	88.31	85.49	91.68	87.51	92.57	90.17	91.07	85.80	93.98
Wine	75.82	66.66	60.84	66.30	71.37	69.05	73.03	68.56	74.77	63.67	70.81
Yeast	51.47	45.20	44.66	46.29	46.62	45.01	45.14	45.89	45.56	45.34	46.42
<b>Average</b>	80.36	65.21	71.27	67.56	74.32	69.30	75.31	71.93	75.08	69.31	74.89

In table 2 we report the results obtained applying DROPs, CLU and OSC over the ten datasets. For each method we show the classification accuracy (*Acc*) and the percentage of the original training set that was retained by each method (*Str*), that is,  $\frac{100|S|}{|T|}$ . In addition, we show the classification obtained by the original training set (*Orig.*). The classifier used was  $k$ -NN with  $k=3$  (the value for DROP methods reported in [1]) and we used as distance function the Euclidean. At the bottom of each table we show the average result of accuracy and storage obtained by each method and the average accuracy difference (*Average diff*) with respect to the original accuracy.

According to the results shown in table 2, in the average case, the best object selection method was DROP2 which reduced the training set size to a quarter approximately. The classification results obtained by OSC were slightly smaller than those obtained with DROP2-DROP5 but OSC outperformed DROP1 and CLU.

The dataset sizes and runtimes spent by DROP1..DROP5, CLU and OSC are shown in table 3. From these results, we can see that for small datasets the OSC's runtimes were not so high with respect to DROPs' meanwhile for large datasets OSC was faster than the DROPs.



**Table 2.** Classification (*Acc*) and retention (*Str*) results obtained by: the original training set (*Orig.*), *DROP1...DROP5*, *CLU* and *OSC*

Dataset	Orig.		DROP1		DROP2		DROP3		DROP4		DROP5		CLU		OSC	
	Acc	Str	Acc	Str	Acc	Str	Acc	Str	Acc	Str	Acc	Str	Acc	Str	Acc	Str
Glass	67.12	100	57.08	23.21	66.73	30.95	63.05	17.70	66.26	27.62	67.71	26.38	63.48	24.00	61.96	39.66
iris	96.00	100	88.00	6.89	96.67	11.63	96.00	10.89	96.00	10.89	95.33	10.74	90.00	17.77	94.66	17.85
liver	64.29	100	67.26	23.45	65.82	34.33	63.47	22.64	66.67	27.99	64.89	26.06	58.24	5.15	58.49	44.02
Pendigitis	99.43	100	90.91	2.66	98.67	5.72	98.89	5.67	98.80	5.70	98.16	4.00	85.74	0.88	94.32	6.23
Segmenta- tion	95.10	100	83.10	12.17	93.24	17.89	92.19	15.94	92.81	17.24	91.86	14.30	70.57	2.22	84.76	15.14
Sonar	82.71	100	68.74	29.65	75.95	33.28	74.60	30.13	76.02	32.32	75.95	29.86	56.73	8.54	79.76	35.41
Spambase	80.55	100	74.37	17.98	78.68	24.69	78.44	15.71	79.09	20.04	78.72	20.97	66.95	0.28	71.95	24.74
Thyroid	93.98	100	86.47	11.63	93.01	17.78	90.76	12.15	92.10	14.32	90.28	12.71	90.17	12.40	91.50	15.03
Wine	74.12	100	70.78	18.41	71.37	23.53	71.96	14.92	71.96	19.41	68.50	19.23	68.56	14.98	71.56	35.39
Yeast	54.64	100	54.04	23.93	56.53	32.93	56.26	19.54	56.19	26.97	54.58	25.30	46.96	4.49	53.04	43.20
<b>Average</b>	80.79	100	74.08	17.00	79.67	23.27	78.56	16.53	79.59	20.25	78.60	18.96	69.74	9.07	76.20	27.67
<b>Average diff</b>			-6.72		-1.13		-2.23		-1.20		-2.20		-11.05		-4.59	

**Table 3.** Datasets sizes and runtimes (in seconds) spent by: *DROP1...DROP5*, *CLU* and *OSC* methods for the results shown in table 2

Dataset	Size		Runtimes						
	Objects	Features	DROP1	DROP2	DROP3	DROP4	DROP5	CLU	OSC
Iris	150	4	0.11	0.19	0.19	0.19	0.16	0.74	0.79
Wine	178	13	0.46	0.82	0.84	0.84	0.59	1.62	1.82
Sonar	208	60	2.14	1.71	1.77	1.79	2.57	1.80	1.94
Glass	214	9	0.40	0.35	0.46	0.40	0.47	0.43	0.54
Thyroid	215	5	0.30	0.32	0.37	0.35	0.40	0.67	0.69
Liver	345	6	0.65	0.50	0.76	0.69	0.76	1.62	1.70
Yeast	1484	8	19.8	14.7	25.6	21.2	23.8	3.97	4.35
Segmentation	2100	19	110.55	205.88	208.39	208.48	208.91	15.84	16.52
Spambase	4601	57	1980.20	2280.62	3782.57	3420.98	2226.42	181.73	189.57
Pendigitis	7494	16	1260.32	2138.64	3180.28	2220.19	2460.05	93.13	94.64
<b>Average Runtimes</b>			337.49	464.37	720.12	587.51	492.41	30.16	31.26

The best methods in table 2 were DROP2...DROP5 since they have been proposed following the  $k$ -NN rule, but it is important to test the object sets (obtained in the previous experiment by DROPs, CLU and OSC) as training sets for other classifiers different from  $k$ -NN in order to know their performance.

By the explained above, another experiment was done using the object sets obtained by DROPs, CLU and OSC as training sets for Locally Weighted Regression (*LWR*, another instance based classifier) and Support Vector Machines (*SVM*, a non instance based classifier). The results are reported in tables 4-5. Based on these results, in the average case, the best object selection method was OSC. Also, we can observe that in the average case, the difference in the

**Table 4.** Classification results obtained using the original training set (*Orig.*) and the object sets obtained by *DROPs*, *CLU* and *OSC* as training for the *LWR* classifier

Dataset	Orig.	DROP1	DROP2	DROP3	DROP4	DROP5	CLU	OSC
Glass	58.76	50.00	48.48	50.09	52.90	52.38	46.70	56.06
Iris	98.00	89.66	93.33	92.00	92.00	92.00	88.00	96.00
Liver	70.13	70.49	70.41	68.26	71.94	69.54	48.39	68.57
Pendigitis	59.63	34.82	46.79	51.66	50.18	54.06	42.40	48.83
Segmentation	78.28	35.76	58.04	55.14	53.80	49.80	34.76	53.33
Sonar	64.40	64.45	65.90	64.50	56.83	59.66	52.23	64.47
Spambase	92.67	88.19	89.82	88.08	89.04	89.76	62.60	89.34
Thyroid	91.16	69.43	75.43	75.43	75.88	75.93	66.51	71.42
Wine	92.15	55.65	68.56	62.75	66.47	60.78	53.88	88.72
Yeast	37.12	35.57	36.99	36.65	36.72	36.86	31.80	35.23
<b>Average</b>	<b>74.23</b>	<b>59.45</b>	<b>65.38</b>	<b>64.46</b>	<b>64.58</b>	<b>64.08</b>	<b>52.73</b>	<b>67.20</b>
<b>Average diff</b>		<b>-14.78</b>	<b>-8.85</b>	<b>-9.77</b>	<b>-9.65</b>	<b>-10.15</b>	<b>-21.50</b>	<b>-7.03</b>

**Table 5.** Classification results obtained using the original training set (*Orig.*) and the object sets obtained by *DROPs*, *CLU* and *OSC* as training for the *SVM* classifier

Dataset	Orig.	DROP1	DROP2	DROP3	DROP4	DROP5	CLU	OSC
Glass	72.29	59.09	63.48	59.74	62.16	63.50	56.03	65.84
Iris	96.66	85.33	92.00	91.33	92.00	93.33	85.33	93.33
Liver	70.72	58.55	58.55	58.26	57.97	56.78	52.50	64.07
Pendigitis	99.65	94.91	98.45	98.89	98.99	98.42	92.06	98.49
Segmentation	91.58	85.28	86.80	86.80	87.47	85.38	70.95	83.95
Sonar	85.14	62.57	66.40	65.35	67.73	65.90	61.02	71.21
Spambase	93.30	91.08	91.06	90.56	90.91	92.02	57.17	88.43
Thyroid	93.54	80.47	90.47	90.45	90.75	93.03	87.01	90.78
Wine	97.18	90.98	94.44	94.93	93.30	92.71	74.21	95.52
Yeast	57.81	52.78	56.37	50.60	54.17	56.81	47.29	54.55
<b>Average</b>	<b>85.79</b>	<b>76.11</b>	<b>79.80</b>	<b>78.69</b>	<b>79.55</b>	<b>79.79</b>	<b>68.36</b>	<b>80.62</b>
<b>Average diff</b>		<b>-9.67</b>	<b>-5.98</b>	<b>-7.10</b>	<b>-6.24</b>	<b>-6.00</b>	<b>-17.43</b>	<b>-5.17</b>

classification accuracy between OSC and DROP2 (the best DROP method obtained in our experiments) using *k-NN* (table 2) is lower than the obtained using *LWR* and *SVM* (tables 4 and 5) where OSC was the best method.

## 5 Conclusions

In this paper, we have proposed, tested and compared the OSC method based on clustering for object selection. This method finds border objects since through these objects it is possible to preserve discrimination capability between classes in a training sample.

The experimental results showed that OSC is a good method for solving the object selection problem when a classifier different from *k-NN* is used. Since

most of the object selection methods follow the nearest neighbor rule, object sets obtained by these methods have a good performance using  $k$ - $NN$ , as it can be seen in our experimental results. However, our experiments showed that these sets did not have a good performance for the  $LWR$  and  $SVM$  classifiers as  $OSC$  did.

As future work, we will do experiments with more data sets and more classifiers. In addition, we will adapt our  $OSC$  method to be applied to mixed data, that is, quantitative (numeric) and qualitative (non numeric) features because in practice it is very common to face problems described by these kinds of features.

## References

1. Wilson, D. Randall and Martínez, Tony R. (2000). Reduction Techniques for Instance-Based Learning Algorithms. *Machine Learning* 38:257–286.
2. Brighton, H. and Mellish, C. (2002). Advances in Instance Selection for Instance-Based Learning Algorithms. *Data Mining and Knowledge Discovery* 6:153–172.
3. Hart, P. E. (1968). The Condensed Nearest Neighbor Rule. *IEEE Transactions on Information Theory* 14(3):515–516.
4. Wilson, D. L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man, and Cybernetics* 2(3):408–421.
5. Tomek, I. (1976). An Experiment with the Edited Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man, and Cybernetics* 6(6):448–452.
6. Lumini, A. and Nanni, L. (2006). A clustering method for automatic biometric template selection. *Pattern Recognition* 39:495–497.
7. Blake, C., Keogh, E., Merz, C. (1998). UCI repository of machine learning databases [<http://www.ics.uci.edu/mllearn/MLRepository.html>]. Department of Information and Computer Science, University of California, Irvine, CA.

---

# Multi-directional Multi-resolution Transforms for Zoom-Endoscopy Image Classification

Roland Kwitt and Andreas Uhl

Department of Computer Sciences  
University of Salzburg  
{rkwitt,uhl}@cosy.sbg.ac.at

**Summary.** In this paper, we evaluate the discriminative power of image features, extracted from subbands of the Gabor Wavelet Transform and the Dual-Tree Complex Wavelet Transform for the classification of zoom-endoscopy images. Further, we incorporate color channel information into the classification process and show, that this leads to superior classification results, compared to luminance-channel based image processing.

## 1 Introduction

Recent statistics of the American Cancer Society reveal that colorectal cancer is the third most common cancer in men and women and the second most common cause of US cancer deaths. Since most colorectal cancers develop from polyps, a regular inspection of the colon is recommended, in order to detect lesions with a malignant potential or early cancer. A common medical procedure to examine the inside of the colon is colonoscopy, which is usually carried out with a conventional video-endoscope. A diagnostic benefit can be achieved by employing so called zoom-endoscopes, which achieve a magnification factor of up to 150 by means of an individually adjustable lens. In combination with dye-spraying to enhance the visual appearance (chromo-endoscopy) of the colon mucosa, zoom-endoscopy can reveal characteristic surface patterns, which can be interpreted by experienced physicians. Commonly used dyes are either methylene-blue, or indigo-carmin, which both lead to a plastic effect. In the research work of Kudo et al. [1], the macroscopic appearance of colorectal polyps is systematically described and results in the so called *pit-pattern* classification scheme.

In this work, we evaluate the discriminative power of features extracted from subbands of the Gabor Wavelet Transform and Kingsbury's Dual-Tree Complex Wavelet Transform, to allow computer-assisted pit-pattern classification. Existing approaches in this research area include histogram- and 2-D DWT-based methods for pit-pattern classification [5, 6] or classic video-endoscopy image classification by statistical second-order measures [7, 8].

The paper is structured as follows: in Section 2 we briefly illustrate the pit-pattern classification scheme. Section 3 outlines the proposed feature extraction

step and the classification process. Finally, Section 4 presents our experimental results and Section 5 concludes the paper with an summary of our work.

## 2 Pit-Pattern Classification

Polyps of the colon are a frequent finding and are usually divided into metaplastic, adenomatous and malignant. Since the resection of all polyps is rather time-consuming, it is imperative that those polyps which warrant resection can be distinguished. Furthermore, polypectomy<sup>1</sup> of metaplastic lesions is unnecessary and removal of invasive cancer may be hazardous. The classification scheme of Kudo et al. divides the mucosal crypt patterns into five groups (pit-patterns I-V, see Figure 1). While types I and II are characteristic of benign lesions and represent normal colon mucosa or hyperplastic polyps, types III-V represent neoplastic, adenomatous and carcinomatous structures. Our classification problem can be stated as follows: the problem to differentiate pit-types I and II from III-V will be denoted as the *two-class* problem, whereas the more detailed discrimination of pit-types I-V will be denoted as the *six-class* problem.

By comparing the prediction of dignity with the histological findings, recent studies [2] have shown, that zoom-endoscopy performs significantly better than the conventional video-endoscopy method, at least in the two-class problem. Although the classification based on Kudo’s results seems to be straightforward at first glance, the interpretation of the specific pit-patterns can be challenging [3]. Furthermore, a high-degree of inter-observer variability, which has been reported at least for diagnosing specialized intestinal metaplasia in Barret’s esophagus [4], can be a disturbing factor. Therefore, our work intends to find a set of image features with enough discriminative power to enable computer-assisted pit-pattern classification.

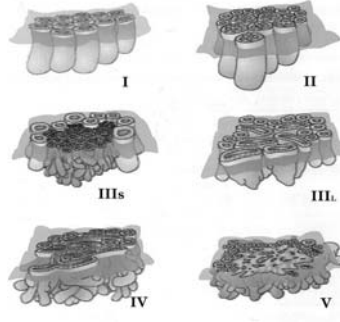
## 3 Feature Extraction and Classification

Each image of our database is decomposed by the Gabor Wavelet Transform [10] and Kingsbury’s Dual-Tree Complex Wavelet Transform (DT-CWT) [9]. Both transforms aim to remedy one commonly known shortcoming of the maximally decimated 2-D tensor-product DWT: the inability to capture directional information, which results from the fact that the wavelet filters are separable and real. With respect to our colonoscopy images, this particular shortcoming is important, since the images exhibit structures that are directed at various orientations (see Figure 1(b)). The mother wavelet in the Gabor Wavelet transform is defined by

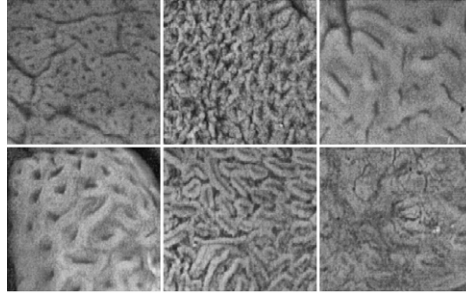
$$\psi(x, y) = \left( \frac{1}{2\pi\sigma_x\sigma_y} \right) \exp \left[ -\frac{1}{2} \left( \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) + 2\pi j W x \right]. \quad (1)$$

---

<sup>1</sup> The process of removing polyps.



(a) Pit-Types



(b) Pit-Types I-V (left to right)

**Fig. 1.** Pit-Pattern classification according to Kudo et al. [1]

The Gabor wavelets for a given number of orientations  $N$  and decomposition scales  $S$  are obtained through the generating function

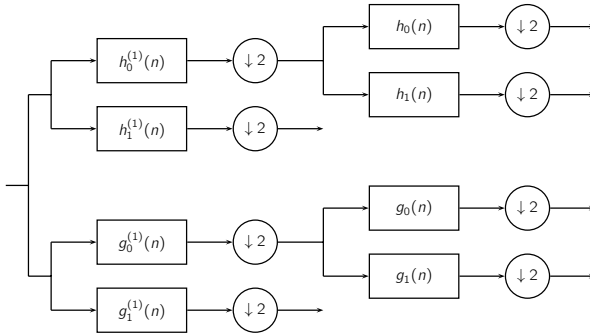
$$\psi_{mn}(x, y) = a^{-m} \psi(\hat{x}, \hat{y}), \quad a > 1, m, n \in \mathbb{N} \quad (2)$$

with  $m = 1, \dots, S - 1$  and scale factor  $a$ . The coordinates  $\hat{x}, \hat{y}$  are given by

$$\hat{x} = a^{-m}(x \cos \theta + y \sin \theta) \text{ and } \hat{y} = a^{-m}(-x \sin \theta + y \cos \theta). \quad (3)$$

Here,  $\theta = n\pi/N$  ( $n = 1, \dots, N$ ) denotes the rotation angle. To obtain a frequency partitioning, where the half-peak magnitudes of the filters touch each other (see Figure 3), the filter parameters  $\sigma_u = 1/2\pi\sigma_x, \sigma_v = 1/2\pi\sigma_y$  in the frequency domain are related to each other according to [10]. The final decision is, how to choose the upper  $U_u$  and lower  $U_l$  center frequencies of the filters, which, together with the number of decomposition scales  $S$ , determine the scaling factor  $a$ . The second transformation we employ here, the DT-CWT, provides an efficient realization of a complex wavelet transform, which is approximately shift-invariant and produces approximately analytic wavelets, oriented at a fixed

number of six directions ( $\approx 15^\circ, 45^\circ, 75^\circ, 105^\circ, 135^\circ, 165^\circ$ ). Furthermore, the DT-CWT is only four-times redundant in 2-D and allows an efficient implementation based on parallel filter bank trees. The key concept is, that the filters in each tree are designed so that the real wavelets they generate from an approximate Hilbert transform pair and the resulting complex wavelet is thus approximately analytic. The basic structure of the DT-CWT in 1-D is shown in Figure 2, where a different set of low- ( $h_0(n), g_0(n)$ ) and high-pass ( $h_1(n), g_1(n)$ ) filters is used in the first stage. This is a crucial requirement of the DT-CWT, because otherwise the first stages would not be approximately analytic [13]. One tree of the DT-CWT gives the real part of the complex wavelet coefficients, while the other gives the imaginary part. The extension to two dimensions is straightforward and requires four separable wavelet transforms in parallel.



**Fig. 2.** DT-CWT analysis FB (1-D)

Our feature extraction step is based on the assumption, that the marginal distributions of the transform coefficients  $\{x_{in}\}_{1 \leq n \leq L_i}$ ,  $x_{in} \in \mathbb{C}$  of subband  $i$  characterize the various pit-types. Here,  $L_i$  denotes the number of coefficients of subband  $i$ . We use the classic mean  $\mu_i$  and standard deviation  $\sigma_i$  of the magnitude of the transform coefficients, given by

$$\mu_i = \frac{1}{L_i} \sum_{j=1}^{L_i} |x_{ij}| \quad \sigma_i = \frac{1}{L_i} \left( \sum_{j=1}^{L_i} (|x_{ij}| - \mu_i)^2 \right)^{\frac{1}{2}} \quad (4)$$

to constitute a general feature vector  $\mathbf{v} = [\mu_1, \sigma_1, \dots, \mu_B, \sigma_B]^T$  for a given image, with  $B$  denoting the total number of subbands. The same image features have already been successfully used in [10, 11] for texture image retrieval.

To incorporate color information into the classification process, we first decompose each image into its red (R), green (G) and blue (B) color channel and transform each channel separately. Thus, we obtain three features vectors  $\mathbf{v}_R, \mathbf{v}_G$  and  $\mathbf{v}_B$  for each image of our database. To show, that color channel information can improve classification results, we also compute a feature vector  $\mathbf{v}_L$  for the luminance (L) channel.

A question, that has not been answered yet is, how to choose the parameters of the Gabor Wavelet Transform and the DT-CWT. Regarding the Gabor Wavelets, we have to choose  $U_l$  and  $U_u$ , as well as the number of decomposition scales  $S$  and orientations  $N$ . In [10], the authors found that four scales and six orientations, with  $(U_l, U_u) = (0.05, 0.4)$  was optimal for their problem. This led to a scaling factor of  $a = 2$ , where the center frequencies are one octave apart. In our studies, we adopt the number of orientations and scales, but we vary the upper and lower center frequencies between  $0.02 \leq U_l \leq 0.08$  and  $0.1 \leq U_u \leq 0.4$  for each color channel. Our results show, that this is a reasonable step, since it is doubtful that the same parameter setting is optimal for all channels. In case of the DT-CWT the number of orientations is fixed to six and we choose a decomposition depth of four scales. Thus, we obtain a total of  $B = 24$  subbands for each color channel for both transforms. For that reason, the dimensionality of the resulting feature vectors is  $|\mathbf{v}_R| = |\mathbf{v}_G| = |\mathbf{v}_B| = 48$ .

Since we either get three feature vectors for the RGB image or one feature vector for the luminance channel, the question arises, how to combine the color information to obtain the best classification results. A first intuitive approach is to *concatenate* the feature vectors of the color channels for each image into one big feature vector  $\mathbf{v} = [\mathbf{v}_R \mathbf{v}_G \mathbf{v}_B]^T$  with  $3 \cdot 48 = 144$  dimensions.

Another approach to tackle this problem is, to use a so called *multi-classifier*, with three single classifiers operating on the distinct feature sets. The outputs of the classifiers are then combined by using the common combining rules, introduced by Kittler et al.[12]. In multi-classifier terminology, the resulting classifier is a so called *parallel multi-classifier*. We will use k-NN classifiers as building blocks for our experiments. To obtain a reasonable error prediction, we employ leave-one-out cross-validation (LOO-CV). The LOO-CV results will be given by the classification *accuracy*, which is defined as the number of correctly classified samples (true positives) divided by the total sample size.

## 4 Experimental Results

In this section, we present the experimental results of our work. Our image database consists of 484 images acquired in 2005/2006 at the Department of Gastroenterology and Hepatology (Medical University of Vienna) using a zoom-endoscope (Olympus Evis Exera CF-Q160ZI/L) with a magnification factor of 150. To enhance visual appearance, dye-spraying with indigo-carmin was applied and biopsies or mucosal resections were taken to obtain a histopathological diagnosis. For pit-pattern types I,II and V, biopsies were taken, since these types

**Table 1.** Number of image samples per pit-pattern class (ground truth)

I	II	III-L	III-S	IV	V
126	72	62	18	146	60



**Table 2.** Comparison of the two- and six-class LOO-CV results for both transforms, using a k-NN classifier with  $k = 1$ . The top results for each color channel are marked bold.

Pit-Type	I	II	III-L	III-S	IV	V	Total
<b>Gabor Wavelet Transform</b>							
2-cl., red	85.86		90.21				88.43
2-cl., blue	85.35		88.81				<b>87.40</b>
2-cl., green	83.33		89.86				87.19
6-cl., red	73.81	61.11	61.29	38.89	69.18	63.33	66.32
6-cl., blue	73.02	61.11	61.29	38.89	68.49	60.00	65.50
6-cl., green	65.08	58.33	79.03	50.00	72.60	60.00	<b>66.94</b>
<b>Dual-Tree Complex Wavelet Transform</b>							
2-cl., red	90.91		95.10				<b>93.39</b>
2-cl., blue	84.34		87.76				86.36
2-cl., green	86.87		89.51				<b>88.43</b>
6-cl., red	80.95	73.61	70.97	72.22	79.45	75.00	<b>77.07</b>
6-cl., blue	74.60	63.89	69.35	44.44	69.86	71.67	<b>69.42</b>
6-cl., green	65.87	62.50	61.29	55.56	65.07	63.33	63.84

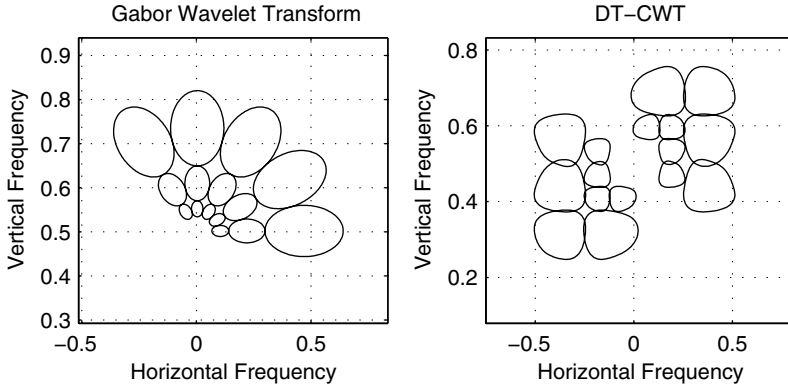
**Table 3.** Optimal parameter settings for the Gabor Wavelet Transform

	2-class				6-class			
	Gray	Red	Blue	Green	Gray	Red	Blue	Green
$U_l$	0.05	0.05	0.07	0.08	0.05	0.06	0.07	0.05
$U_u$	0.40	0.50	0.30	0.40	0.30	0.40	0.20	0.30
$\rightarrow a$	2.00	2.15	1.62	1.71	1.81	1.88	1.42	1.81

need not be removed. Lesions of pit-pattern types III(S/L) and IV have been removed endoscopically. Table 1 lists the number of image samples per class.

Before actually transforming the images, we perform two image-quality enhancing preprocessing steps: histogram equalization using the CLAHE [14] (contrast limited histogram equalization) algorithm and blurring with a  $3 \times 3$  Gaussian kernel ( $\sigma = 0.5$ ). The LOO-CV results for the red, green and blue color channel are given in Table 2. The numbers represent the classification accuracy of the LOO-CV.

We can see, that for both transforms the features obtained from the red color channel lead to the best results in the two-class problem, whereas the features from the green color channel are most discriminative in the six-class problem. Regarding the parameter  $k$  for the k-NN classifier, we ran several tests with  $1 \leq k \leq 20$ , with  $k = 1$  leading to the best results in almost all cases. Table 3 lists the best parameter settings ( $U_l, U_u$ ) for the Gabor Wavelet transform, together with the resulting scale factor  $a$ . We point out, that the optimal parameters for the



**Fig. 3.** Frequency partitioning for the Gabor Wavelet Transform and the DT-CWT

color channels are in no case equal to the parameters in [10]  $(U_l, U_u) = (0.05, 0.4)$  and additionally differ for the two classification problems.

To illustrate the difference between the Gabor Wavelet Transform and the DT-CWT regarding the frequency partitioning, Fig. 3 shows the half-peak (Gabor Wavelets) and 70% peak magnitude (DT-CWT) of the filter responses in the frequency domain for a selection of filters from both transforms. The parameters of the Gabor Wavelet Transform are the optimal parameters for the red color channel in the two-class problem.

Regarding the combination of the information from the different color channels of the RGB color model, Table 4 lists the LOO-CV results for both transforms and combination methods, together with the LOO-CV results of the luminance (gray) channel features. The concatenation scheme is abbreviated by *concat.* and the multi-classifier is abbreviated by *MC*.

For both transforms, the so called *product rule* [12] performs best for the two- and six-class problem using the multi-classifier. Although, the overall best LOO-CV results are obtained from the DT-CWT features using the simple concatenation scheme. Due to the fact, that no combination scheme is optimal in all cases, we follow that there is no common rule for combining color channel information for our classification problem. However, both combining schemes are at least superior to the luminance channel features alone.

## 5 Conclusion

In this work, we presented an approach to computer-assisted pit-pattern classification by means of two multi-directional multi-resolution transforms. Compared to the top results in [6] (75% in the two-class problem and 57% in the six-class problem), which are based on the same image database, we have come to the conclusion that the directional selectivity of both transforms leads to improved classification results and outperforms feature extraction based on the maximally

**Table 4.** Comparison of the LOO-CV results for the luminance channel and the combined color channel information, using simple concatenation and a 1-NN multi-classifier with the *product* combining rule. The top results for both classification problems are marked bold.

Pit-Type	I	II	III-L	III-S	IV	V	Total
<b>Gabor Wavelet Transform</b>							
2-cls., gray	87.37		89.86				88.44
2-cls., concat.	88.89		92.66				91.12
2-cls., MC	89.90		95.45				93.18
6-cls., gray	73.63	63.89	62.90	66.67	74.66	56.67	69.98
6-cls., concat.	77.78	66.67	74.19	50.00	71.23	66.67	71.28
6-cls., MC	82.54	68.06	75.81	38.89	84.93	63.33	
<b>Dual-Tree Complex Wavelet Transform</b>							
2-cls., gray	87.88		91.61				90.08
2-cls., concat.	92.83		98.60				<b>98.28</b>
2-cls., MC	93.43		97.90				96.07
6-cls., gray	73.02	68.06	66.13	55.56	69.86	70.00	69.42
6-cls., concat.	85.71	69.44	88.71	83.33	82.88	78.33	<b>81.82</b>
6-cls., MC	84.92	68.06	82.26	66.67	84.25	76.67	80.17

decimated 2-D DWT. In addition, color channel information leads to even better results and is superior to luminance-channel based image processing, at least for our two classification problems. This is consistent with the results in [5], where information from three-dimensional color histograms led to the top classification results. Regarding the choice, which transform to choose, we favour the DT-CWT, since it is less computationally expansive than the Gabor Wavelet transform and leads to slightly better classification results.

## References

1. Kudo, S. et al. (1994) Colorectal tumorous and pit pattern. *Journal of Clinical Pathology* 47:880–885
2. Konishi K., Kaneko K. et al. (2003) A comparison of magnifying and nonmagnifying colonoscopy for diagnosis of colorectal polyps. *Gastrointestinal Endoscopy*, 57:48–53
3. Hurlstone, D.P. (2002) High-Resolution magnification chromoendoscopy: Common problems encountered in pit-pattern interpretation and correct classification of flat colorectal lesions. *American Journal of Gastroenterology*, 97:1069–1070
4. Meining, A. et al. (2004) Inter- and intra-observer variability of magnification chromoendoscopy for detecting specialized intestinal metaplasia at the gastroesophageal junction. *Endoscopy*, 36:160–164
5. Hfner, M., Kendlbacher, C. et al. (2006) Pit Pattern Classification of Zoom-Endoscopic Colon Images Using Histogram Techniques. In: *Proceedings of the 7th Nordic Signal Processing Symposium*, 58–61

6. Hfner, M., Liedlgruber, M., et al. (2006) Pit Pattern Classification of Zoom-Endoscopic Colon Images Using Wavelet Texture Features. In: 3rd International Conference MEDSIP: Advances in Medical, Signal and Information Processing, pp.37
7. Maroulis D. et al. (2003) CoLD: A versatile detection system for colorectal lesions in endoscopy video frames. In: Computer Methods and Programs in Biomedicine, 70: 151–186
8. Karkanis, S.A., Maroulis D. et al. (2003) Computer-Aided Tumor Detection in Endoscopic Video Using Color Wavelet Features. IEEE Transactions on Information Technology in Biomedicine, 7(3):141–152
9. Kingsbury N. (1998) The dual-tree complex wavelet transform: A new technique for shift-invariance and directional filters. In: Proceedings of the 8th IEEE DSP Workshop, pp. 9–12
10. Manjunath B.S., Ma. W.Y. (1996) Texture features for browsing and retrieval of image data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(8):837–842
11. Rivaz P., Kingsbury, N. (1999) Complex wavelet features for fast texture image retrieval In: Proceedings of the IEEE Conference on Image Processing, pp. 109–113
12. Kittler, J. et al. (1998) On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3):226–239
13. Selesnick, I.W., Baraniuk, R.G., Kingsbury, N. (2005) The Dual-Tree Complex Wavelet Transform. IEEE Signal Processing Magazine, 22(6):123–151
14. Zuiderveld, K. (2004) Contrast Limited Adaptive Histogram Equalization. Graphics GEMS IV, pp. 474–484

---

# Image Retrieval for Image Theft Detection

Ondřej Horáček, Jakub Bican, Jan Kamenický, and Jan Flusser

Institute of Information Theory and Automation, Academy of Sciences of the  
Czech Republic,  
Pod Vodárenskou věží 4, 182 08 Prague 8, Czech Republic  
{horacek,bican,kamenik,flusser}@utia.cas.cz

**Summary.** Image retrieval deals with a problem of finding similar pictures in image database. Our task is to find originals of modified images, typically stolen and republished on the web. Our problem is specific in terms of the database size (millions of photos), demanded speed of the search (seconds), and unknown image modifications (loss of quality, radiometric degradation, crop, etc.). Proposed method works in the following tree steps: 1. Image preprocess – normalization for robustness to the modifications. 2. Retrieval of candidates from the database index – stochastic decision in each vertex of the index tree is used to find the most relevant candidates. 3. Verification of the candidates – modified phase correlation is used. The method was implemented in practice with very good results. Based on wide experiments, it was shown that the success rate of the search depends on the level of image modification.

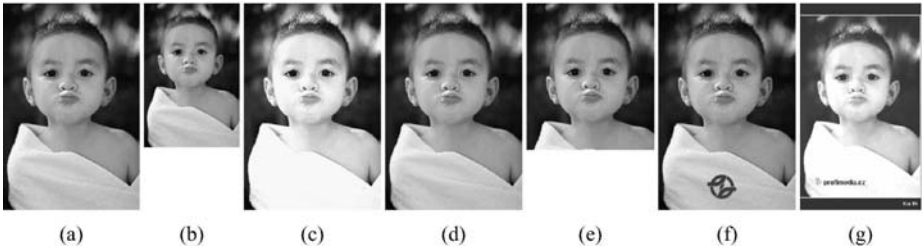
## 1 Introduction

Large image databases are often run on a commercial basis – browsing through and viewing images is free of charge while downloading and re-using them on your webpages and articles is a subject of a fee. However, some users republish the downloaded images without paying the fee, which is a violation of copyright law. The copyright owner thus wants to regularly scan suspicious domains or websites to check if there are any unauthorized copies of the database images.

This paper describes an original method which we developed for an international advertising and press company. This company runs a database of more than 10 millions photographs updated everyday. They estimate hundreds thousands images being used without permission on the web. Detection of illegal copies is complicated by two principal difficulties – the unauthorized images are usually modified before they are post on the web and the response of the system must be extremely fast because of an enormous number of database images. Although this problem formulation looks like an image retrieval task, this is not the case. In traditional image retrieval, we want to find in the database all *similar* images to the query image, where similarity is evaluated by colors, textures, content, etc. Here we want to identify only the *equivalent* images to the query (we

call this task *image identification*. This is why we cannot apply most of standard image retrieval techniques. By the term "equivalent images" we understand any pair of images which differ from one another by the following transformations.

- Quality reduction. Either compression to different image format or resize changes the image representation, although the image seem very similar to human eye (in Fig. 1b).
- Radiometric and color distortions. We consider changes of image brightness and contrast (in Fig. 1c), changes of color tone or conversion of the image to gray-scale (in Fig. 1d).
- Crop of the image. Image part can be cropped from the background, still we consider that the major part of the image is preserved. Also, a frame can be added to the image or aspect ratio can be changed (in Fig. 1e).
- Local changes. A logo can be added to the image, or a thin label can go throw the image (in Fig. 1f).
- Combinations. Reasonable combination of distortions mentioned above is also considered. However, their increasing amount and significance will surely impact the algorithm results (in Fig. 1g).



**Fig. 1.** Possible modifications of the query image. The first image is the original stored in the database.

We decided to include the above transforms into our "equivalence relation" because, according to the earlier statistics performed by the company, they are frequently present in unauthorized copies.

It is not possible to use directly any of the existing methods. Probably the closest published method is by Obdržálek et al. in [4]. It is suitable for recognition of man-made objects with partially planar surface, such as goods in a supermarket. It can handle very general situations including partial occlusion and affine transform of the objects but this is useless for our purpose. Our method is different in several aspects. We can not use several maximum extremal regions per image, because on our database images they may not exist. Our index tree is thousand times bigger than that one considered in [4], so we need to use different, space-efficient tree structure. Finally, we include another independent image comparison to eliminate false positives matches.

We present an original image identification method, which is based on a hierarchical structure of the database, representation of the images by proper invariant features, and a fast tree-searching algorithm. Our method has got very good identification rate in a reasonable response time. In this article, we present main idea of the method as well as selected details.

## 2 Algorithm Outline

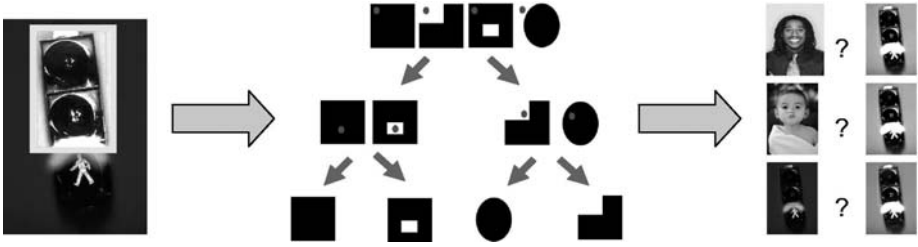
A kind of binary decision tree is used for the database indexing. Some image features are needed to characterize the image in the index. We require the features to be robust to considered degradations, stable, but mainly to be extensible and discriminate enough for any database size. We use image intensities in various pixel positions, surely after dealing with the image distortions. This choice is simple in principle, but we found it effective in presented method. Our image identification works in these three steps:

1. Normalization. Robustness to the modification is ensured by normalization of the images during a process. In other words: Each of the considered modification corresponds with a change of an evaluative image quality. We apply the modification once again in an amount, that was established to set the qualities to the same value for all the images. This process annuls the impact of considered modifications.
2. Stochastic index. The database images are organized in binary decision tree. Thus, we obtain several candidates for match with a query image very quickly. Decisions in the tree are based simply on image intensity at certain position. The position and threshold are set for each vertex during build of tree. For individual query image, stability of decisions in the tree is evaluated. We alternate the unstable decisions during the image identification. So, we get many candidates per query.
3. Candidate verification. Edge information of the image serves for the the final comparison of a candidate with the query image. More concretely, phase correlation restricted to low-pass fourier transform is used.

## 3 Normalization

Both the query image and the database images are preprocessed previously. We apply some normalizations to make the images invariant to considered modifications. First, we convert the images to gray-scale to handle color distortions. Then, the lowest and the highest 10 percent image histogram fraction are found and their centers of gravity are set to fixed values (by addition / multiplication of the image intensity values). This ensures image invariance to brightness and contrast changes.

The image crop is tougher modification to handle. The first idea could be to use features invariant to crop, such as corners (found for example by Harris corner detector [1]) or intersection points. We do not use them because it is not



**Fig. 2.** Schema of the image identification. First, normalization of the query image is done. Then, the most probable candidates are found in the index tree. And finally, the candidates are verified by modified phase correlation.

possible to stably match (identify) those points for all the database images after crop. We consider that the crop preserves major part of the image. For most of the images, the part is separable from the background by color. So, we segment the major part of the image and bound it by a crop invariant frame.

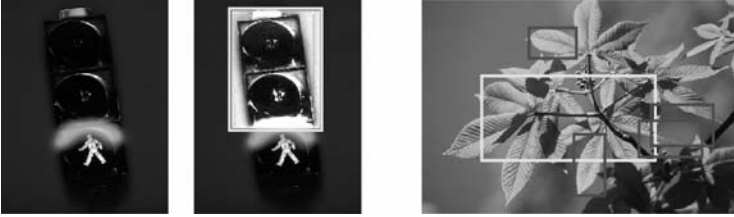
We introduce a special color-based segmentation for separation of the image major part. The algorithm finds a frame in the image which fulfill following requirements: it is big enough, it does not lay on the image border, it is color specific and it as stable as possible. It was developed heuristically with respect to experiment results. Our algorithm works in principle as follows: In principle, it divides the image into blocks, computes the block color character, and finds neighborhood blocks with the same character for each possible starting block. The segmented region is broadly bounded by box, which we call a frame. Stability and "quality" of a the frame is evaluated. For the database images, just area bounded by the best frame is used for image identification in the index tree. It is reasonable to consider that this frame will be found in the modified image as one of the best, too. Therefore we use the best five variants of the frame for the image search.

## 4 Stochastic Index

The task for the index is to retrieve image from the database quickly. Response time to a query must be principally shorter than proportional to the database size, which goes to millions of images. Input for the index is a query image that have been normalized and blurred the same way as the database images were. They should be very similar, but still, we need to evaluate stability of each decision in the index tree to be robust to minor image changes.

The database images are organized in a binary decision tree, commonly used to handle huge amount of data (a survey is done in [3]). Decision in the tree are based on intensities of image pixels. The pixel (threshold) position is taken relatively to the valid image area (frame). Once we have two different branches of the tree, each of them can contain images differing mainly (and therefore the most stably) in a different image part. Such a threshold position is established

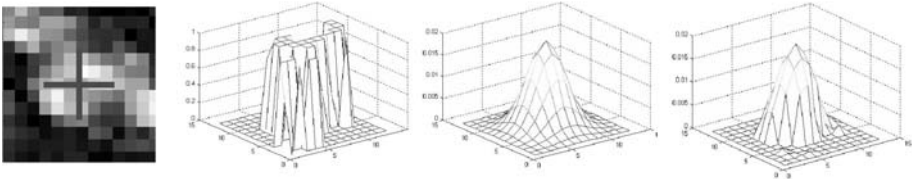




**Fig. 3.** Image preprocess. The original is converted to gray-scale, crop-invariant frame is found and its brightness and contrast are normalized. On the right, multiple invariant frames used for the image identification are shown.

during the tree build. So, each node of the tree contains threshold value and relative position of the threshold pixel.

Now we describe the search for image in the index tree. In a node, pixel of the query image is compared with the threshold value and its sub-branch (next node) is chosen. We establish stability of the comparison as a likelihood, that the image belong to the same (left/right) branch even after following image distortions. We assume that the threshold pixel can change its intensity (with uniform distribution in certain intensity interval) and it can change its position (the miss-place has a two-dimensional gaussian distribution) (in Fig. 4). This stochastic model is similar to the one presented by Obdržálek [4], but their usage of the model is different.



**Fig. 4.** Stability of decision in the tree. The image and the threshold position, likelihood that a pixel is above the threshold, likelihood of threshold position change, and their multiplication.

This way we go through the index tree till we reach a leaf containing an image – candidate for the match. Now, we select node with the least stable decision. Now we continue through the alternate branch and get another match candidate. This way we found 20 images from the index with the highest probability to match the query image. Note, that some candidates are found even for totally dissimilar query image.

It was mentioned above that the index tree needs to be prepared previously. Because of the huge number of images, the tree is built gradually. Our algorithm works in principle as follows:

1. Read normalized version of a database image from a disk one by one.
2. Find a leaf in the partially builded tree for the image and register the image to the leaf.
3. If the leaf contains enough images, convert the leaf to inner tree node and redistribute its images as follows:
  - a) Choose randomly several positions of a threshold.
  - b) For each threshold candidate, get intensity values at the processed position (for all the images belonging to the leaf).
  - c) For the threshold candidate, compute the threshold value as a median of the intensity values.
  - d) For the threshold candidate, compute its stability as a sum of decision stabilities for all the images belonging to the leaf. It means, evaluate the likelihood described in the paragraph about search (and in Fig. 4).
  - e) Choose the most stable threshold of the threshold candidates and save its position and the processed node (leaf).
  - f) Create left and right sub-node of the processed node. Based on the new threshold, redistribute the images of the processed node to them.
4. After all the database images have been processed by previous steps and are registered to some leaf, divide the leafs till they contain only one image. (Do it the same way as in steps 3a to 3f).

## 5 Rest of the Method

In the last step, the candidate images are compared one by one with the query image. We use modified phase correlation (originally introduced by Kuglin [2] in 1975). Phase correlation is robust to overlap, shift and radiometric degradation. We restrict the correlation only to low-pass of the fourier spectrum to make the comparison more stable for image quality changes (in Fig. 5).

For better performance in practical experiments, many improvements have been made on the basic method scope described above. The algorithm use several parameters and options, that affect both the identification rate and algorithm speed. There is another tradeoff with the normalization: the more quantities are normalized, the more information is lost and it is harder to identify the original image; but it makes it possible to identify images with harder combination of distortions. We overcome this situation by usage of more than one versions of preprocess (with different normalization and different parameter set). And, of course, more than one index tree. We use two independent index trees in the prototype implementation: one is build from images with normalized histogram, the second is build and searches for the image parts bounded by the invariant frames.

The method is being implemented as a configurable framework. A structured configuration file controls application of zero to several normalization algorithms per image, build and search in several index trees as well as it contains all parameters for all the algorithms. (Note, that some simple preprocess algorithms, such as mirroring or blurring, are not described in this article). We have also



**Fig. 5.** Candidate verification. Low-pass of phase correlation bases the image comparison on their major edges.

implemented implemented optional index ability to work with color images. In that case, a threshold value is replaced by a plain in RGB color space.

## 6 Experiment Results

Prototype of the proposed method was implemented in Matlab. Tests have been done on 100 000 image database. First, a thousand of query images was generated for each considered modification (the images are still "equivalent", see samples on Fig. 1). Strength of the modification is varying around the level expected from the practice. The identification ratio is very good, better than we expected. Original images are found successfully in 99.5 % of cases. The rest 0.5 % are cases, when the database contained a very-similar image (e.g. with some retouching), that has been identified before the original one. For typical automatically republished images, identification rate is more than 90 %, which is very good for practical use of the method. The identification rate surely decrease for harder modifications, but even for combinations of radiometric degradations, crop and logo is still about 20%. See table 1 for more details.

Response speed of the Matlab prototype is up to 20 seconds per image. Image retrieval from the index tree takes about 0.2 second, rest belong to the one by one candidate verification by the fourier transform. Build of the index tree takes

**Table 1.** Identification rate on 100000-image database. The identification rate depends on kind of image modification.

Degradation	True positives	False positives
Original	99.5 %	0.4 %
Logo added	98.2 %	0.2 %
Scale	94.6 %	0.4 %
Brightness and contrast	71.2 %	0.7 %
Crop	45.0 %	0.2 %
Scale + logo	93.4 %	0.2 %
Scale + logo + frame	35.8 %	0.4 %
Radiometric deg. + crop + logo	18.2 %	0.5 %
Not in the database	0.0 %	0.3 %

less than a second per image. But, the database build can take several days with some non-trivial normalization (such as invariant frame – about 3 seconds per image). Overall, the method speed as well as the identification rate depends on appropriate set of parameters.

## 7 Conclusion

In this article, we presented our method for modified image identification. The task is specific by character of the modifications, the database size, and required response speed. The method is novel in normalization during the image preprocessing (invariant frame, normalization) and in stochastic backtracking through the image index. It was shown in experiments on huge database that the method performs very well. It is ready to catch majority of illegally republished database images on scanned web sites.

**Acknowledgement.** Ondřej Horáček and Jan Kamenický were supported by the Czech Ministry of Education under the project No. 1M6798555601 (Research Center DAR).

This article was supported by Czech Technical University in Prague under the project No. CTU 0713514.

## References

1. C. Harris and M. J. Stephens. A combined corner and edge detector. In Alvey Conference, pages 147–152, 1988.
2. C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. Assorted Conferences and Workshops, pages 163–165, September 1975.
3. Sreerama K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Min. Knowl. Discov*, 2(4):345–389, 1998.
4. Obdržálek, Štěpán and Matas, Jiří. Sub-linear indexing for large scale object recognition. In WF Clocksin, AW Fitzgibbon, and PHS Torr, editors, *BMVC 2005: Proceedings of the 16th British Machine Vision Conference*, volume 1, pages 1–10, London, UK, September 2005. BMVA.

---

# Practical Evaluation of the Basic Concepts for Face Localization

Maciej Smiatacz

Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Narutowicza 11/12, 80-952 Gdansk, Poland  
slowhand@eti.pg.gda.pl

**Summary.** Although a large number of face localization methods was proposed in the literature and many of these theoretical ideas seem very attractive, their practical value still needs to be verified. Moreover, while the techniques being described in publications are getting more and more sophisticated, they quite often rely on very straightforward concepts like edge maps. Therefore, the main goal of this work was not to propose another face localization method but to carefully evaluate the basic algorithms – in order to identify their practical drawbacks or hidden potential.

## 1 Introduction

Face recognition [1] was one of the first concepts for application of image analysis and pattern recognition techniques to person identification. Some of the basic algorithms appeared in early 1990's and at the end of that decade we already had dozens of new methods and practical solutions.

Every biometric application based on face recognition must perform the localization of the image fragment that will be further analyzed; running a recognition algorithm on the whole image, containing not only the face but also the background and other objects, cannot provide reliable results.

For the above reasons a lot of face localization methods have been described in the literature [2]. However, these publications usually have very brief experimental sections and do not allow the reader to judge the practical value or possible disadvantages of the methods. From the practitioner's point of view the questions of speed and reliability in changing conditions are crucial but they can only be answered when the algorithms are independently implemented and tested.

There is no doubt that the task of evaluating the face localization and recognition methods became easier since the representative databases of images like FERET [3] or BANCA [4] appeared. Thanks to them researchers are able to prepare quantitative results instead of presenting a set of chosen examples. On the other hand, careful analysis of the particular cases in which the algorithm failed, finding the *reason* for which it failed, can provide much more practical information than studying only the final scores.

In the previous work [5] we described results of the face localization tests performed with the help of our implementation of the Active Shape Models. Although the technique is theoretically sophisticated, it revealed several serious problems in real-life situations. Therefore, this time we decided to test the very basic approaches:

- the method of edge-maps [6], that represents the very early idea that a face can be localized by finding the elliptical contour in a scene;
- the color and shape analysis [7], which assumes that a face is an elliptical object with characteristic skin-like color.

The experiments were performed with the help of the ALiSOT system [8] prepared in our department.

## 2 Edge Maps

The method described in this section is based only on the assumption that the outline of a face is oval and that it can be approximated by an ellipse [6]. This approach requires the precise edge map of the analyzed image, which can be acquired with the help of the well-known Canny’s algorithm [9].

Having calculated the edge map we must pay special attention to the existence of edge intersections. If they represent overlapping objects then they should be removed from the map to preserve the consistency of the face outline, which, in turn, is particularly important in the next step, when similar segments are grouped. Thus, the algorithm analyses pixel configurations in  $3 \times 3$  neighborhoods and removes the intersections. Unfortunately, this causes a risk of dividing some of the correct edges into several parts so the special labeling procedure tries to find the edges represented by many separated segments. It constructs vectors along the different segments around each intersection and then calculates the dot product for every pair of the vectors. The pair with the lowest value is treated as a representation of one partitioned edge, so the segments for which the vectors were created are assigned the same label.

When the map of the labeled segments is ready, another algorithm takes pairs of edges and fits ellipses to them. For most of the faces the ratio of the two ellipse axes falls into particular range and this fact is used as a criterion for choosing the face candidates. The ellipse that covers the highest number of edges or contains only one very strong edge is selected as the face outline. The subsequent steps of the algorithm are illustrated on Fig. 1.

Even after analysis of only three sample images from BANCA database, taken in “controlled” conditions, i.e. with uniform background and good lighting, we were able to identify some important drawbacks of the method.

1. Although segmentation of the first image gave quite good results (Fig. 2), the face outline in the edge map is formed from two segments. One of them contains not only the contour of the face but traces also the nose, eyebrow and neck. In this case the algorithm managed to fit the ellipse correctly but the problem of too long segments is a potential source of serious mistakes.



**Fig. 1.** The method of edge maps: (a) source image, (b) final form of the edge map with candidate ellipses, (c) localized face



**Fig. 2.** Improper contours taken into account during the segmentation

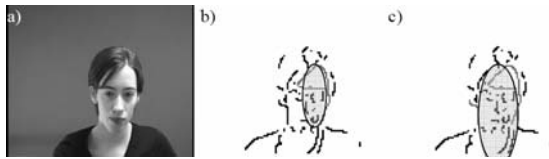
2. In the next image (Fig. 3), the face outline is even more fragmented. This causes troubles because: a) when the ellipse is fitted to the forehead contour, the fitment score is very high, suggesting that this is the best solution, but the face is not segmented accurately (Fig. 3b); b) if we match the ellipse to the two most distinct segments the result appears wrong again as we get too narrow face region (Fig. 3c).



**Fig. 3.** Problems caused by inadequate fitment values

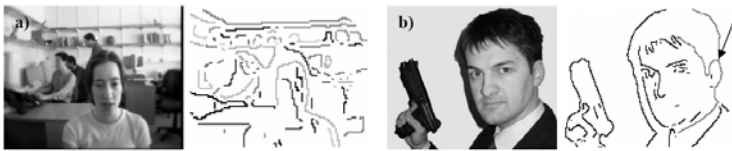
3. The third example (Fig. 4) demonstrates how poor edge map one can obtain even in good, controlled conditions. The longest edge segment represents one side of the face but contains also the horizontal line formed by the eyebrows. If we fit the ellipse to this edge, it will not even cover both eyes of the person (Fig. 4b). We can try to use a pair of edges (Fig. 4c) but then we would receive the low fitment score of the resultant ellipse.

The above critical observations do not mean that the method cannot be useful at all. The image shown on Fig. 1 was segmented almost perfectly despite the low quality of the photograph (taken with the webcam), non-uniform background and adverse lighting. However, in the example shown in Fig. 5a once again we face the problem of one long edge segment that does not allow the ellipse to fit



**Fig. 4.** Poor edge map obtained despite the good conditions

properly. Figure 5b proves that the algorithm removing edge intersections does not always cope with realistic situations – in this case we get a large, branched edge that disturbs the ellipse-finding procedure.



**Fig. 5.** Problems caused by too long edge segments

The final conclusions are as follows.

1. The method uses small amount of information, nevertheless this data should be enough in most of the cases, since the human observer is able to locate the head outline easily even on a very fragmented edge map. This means the general idea is passable, although the original algorithm should be improved in many ways.

2. It is obvious that the method could be practically used only in particular conditions: the proper lighting and reasonable background is essential. The size of the head must fall within some predefined range and only one face can be localized at a time. On the other hand the quality of the image is less important, even the cheapest webcam produces images that are good enough for this method.

3. The main improvements of the method should be related to the proper edge segment extraction and analysis. Not only the intersections but also the points at which the edges are bended too much should be removed from the map. Moreover, the edge labeling and grouping algorithm should become more knowledge-driven; the ellipse matching procedure could be replaced by more advanced curve fitting algorithm. The method should take into account also the orientation of the ellipse so that it could localize not only the upright faces.

4. The analysis of the edge maps obtained for the sample real-life images leads to the observation that in many cases it is not the head outline ellipse that helps humans to locate the face region on the edge map – it is the distinctive and characteristic line representing the neck and shoulders of a person. Therefore, this features should be taken into account during the image analysis, for example they could be included in some form of a face model.

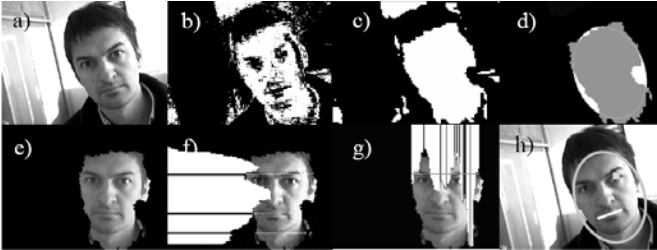


### 3 Color and Shape Analysis

This approach is based on the observation that the human face is characterized by the oval shape and the specific skin color, even in changing illumination conditions [6]. The HSV model is used and the hue and saturation parameters are treated as the ones that carry the most important information. The authors of the algorithm say that this way the human color perception is simulated and the independence from illumination changes can be at least partially achieved. They also indicate that the color thresholding parameters should be set to the following values:  $S_{min} = 0.23$ ,  $S_{max} = 0.68$ ,  $H_{min} = 0^\circ$ ,  $H_{max} = 25^\circ$ .

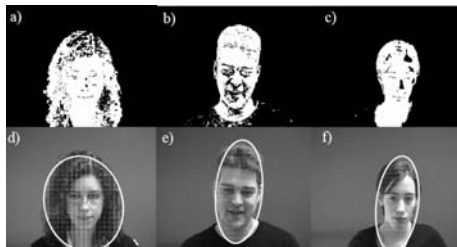
As the oval shape of the face can be approximated by an ellipse, the search for the potential face regions may be reduced to fitting an ellipse to each of the candidate areas. In the first step the region-growing algorithm is employed to find and label all the image fragments with skin-like color. Then for each connected component (within a given size range) the best fitted ellipse is computed on the base of moments. The ellipse is described by its center  $(x, y)$ , the axis  $(a, b)$  and the orientation  $\theta$ . The last parameter must not exceed  $45^\circ$ , and the ratio of major to minor axis should be reasonable (we chose the range of  $[1.0; 2.2]$ ). For each of the candidate ellipses the fitment measure  $V$  is determined. It is based on counting the “holes” inside the ellipse and the points of the connected component that remain outside of the ellipse.

When the head region is found we search it in order to localize the eyes and mouth. This is achieved by analysing the vertical and horizontal grey-level projections extracted from the image. The subsequent steps of the algorithm are illustrated on Fig. 6.



**Fig. 6.** Subsequent steps of the color and shape analysis: (a) source image, (b) skin region, (c) filling holes and smoothing the image, (d) ellipse fitting, (e) normalized face image, (f)  $y$ -projection, (g)  $x$ -projection of the selected horizontal line, (h) final result

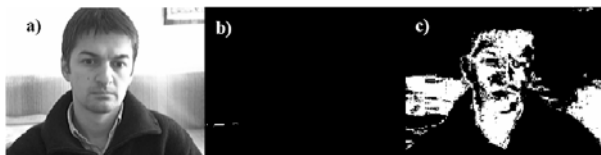
The faces shown on the three examples from the “controlled” subset of the BANCA database were localized passably (Fig. 7). One can notice, however, that in all of the cases the hair and neck were also included in the face region so the ellipses cover too large area. The results are therefore not accurate enough for the face recognition or verification module, because the artifacts included in the ellipse would disturb the process of feature extraction.



**Fig. 7.** (a, b, c) Face regions and (d, e, f) the results of the segmentation

When we tried to perform the segmentation of the images taken with the webcam the results were much worse. This was caused by the poor color reproduction provided by the cheap CCD sensor. For the photograph shown on Fig. 8a we did not obtain any skin region at all (Fig. 8b). The thresholding parameters had to be changed to  $S_{min} = 0.0$ ,  $S_{max} = 0.2$  (very low saturation) and  $H_{min} = 300^\circ$ ,  $H_{max} = 350^\circ$  (purplish-blue skin tones!). This way we managed to extract the face region properly (Fig. 8c) but when we tried to process an image taken a few minutes later with the same webcam (Fig. 9a), the same set of parameters led to extraction of everything but the face (Fig. 9b). We switched to the default values again and the result was different but not better (Fig. 9c). It seems that in the case of the low quality camera and uncontrolled conditions the parameters of the method should be changed almost from frame to frame. This means that the method would not work at all with the popular, Internet-oriented hardware, so it cannot be employed, for example, to control the compression of videoconference images.

Obviously, if the background has the color that belongs to the range of skin tones, the method will fail. The problem is that the colors resembling the skin



**Fig. 8.** Parameters had to be changed to perform the segmentation (see text)



**Fig. 9.** A new set of parameters was needed for the next image

may appear in surprising locations, as we can see on Fig. 10. This is because the intensity of the color is neglected to ensure the independence from illumination changes. As a side effect, however, the face candidates can emerge from the background.



**Fig. 10.** (a) An image taken with a digital camera, (b) skin tones in the background, (c) the real face region was not even taken into account during the ellipse fitting

As we mentioned before, the method of color and shape analysis is also capable of locating the eyes and mouth. However, because in most of the cases the resultant ellipse is too big, the facial feature localization process can be easily disturbed. The typical example is shown on Fig. 11. The upper significant minimum in  $y$ -projection originates from the black eyebrows, not the eyes (Fig. 11b). Unfortunately, the line corresponding to the minimum has the  $x$ -projection that suggests the presence of eyes (Fig. 11c), so the mislead algorithm detects them at a too high position.



**Fig. 11.** The problems with localization of the face features (see text)

We sum up our observations concerning the color and shape analysis as follows.

1. The method is suitable only for very controlled conditions, i.e. ideal illumination and uniform background with the appropriate color. The method works reasonably on high quality images, e.g. the photographs taken with a digital camera but the color misrepresentations introduced by popular devices such as webcams are so significant that the correct localization of the skin region becomes very difficult.

2. The default parameters of the method must be modified very frequently – most often the range of the skin color values must be changed or shifted in order to compensate for the color distortions introduced by the camera.

3. The large background objects in the skin-like color always lead to failure, even if their shape is not elliptical at all.
4. Although the method uses more information than the previous one, the results of face localization are not very accurate even in good conditions.
5. Generally saying, color is *not* the feature that we can depend on when looking for a face in an image.

## 4 Conclusions

Searching for a reliable and *practically useful* method of face localization we implemented the advanced idea of Active Shape Models [5] but the results were disappointing. Therefore, this time (with the help of our ALiSOT system) we tested the very simple concepts, trying to utilize the “less is more” approach. As a consequence we realized that the segmentation based on color is worthless in the case of face images. Our experiments, however, encouraged us to prepare a face localization algorithm based on advanced analysis of edge maps. We will report results of these efforts as soon as the new implementation is ready.

## References

1. Zhao W, Chellapa R, Phillips PJ, Rosenfeld A (2003) Face recognition: a literature survey. *ACM Computing Surveys* 35(4): 399-458
2. Yang M-H, Kriegman D, Ahuja N (2002) Detecting faces in images: a survey, *IEEE Trans Patt Anal Mach Intell* 24(1): 34-58
3. Phillips PJ *et al* (2000) The FERET evaluation methodology for face recognition algorithms. *IEEE Trans Patt Anal Mach Intell* 22(10): 1090-1104
4. Bailly-Baillièrè E *et al* (2003) The BANCA database and evaluation protocol. In *4th Int Conf Audio- and Video-Based Biometric Person Authentication (AVBPA)*: 625-638
5. Smiatacz M, Malina W (2005) Active Shape Models in practice. *Proc 4th Int Conf Comp Rec Systems (CORES'05)*: 451-459
6. Sirohey SA (1993) Human face segmentation and identification. University of Maryland, CS-TR3176
7. Sobottka K, Pitas I (1996) Extraction of facial regions and features using color and shape information. *IEEE Int Conf Patt Rec, Vienna, Austria*: 421-425
8. Kaniewski M (2002) The methods of face localization and tracking (in Polish). MSc Thesis. Gdansk University of Technology, Gdansk, Poland
9. Canny J (1986) A computational approach to edge detection. *IEEE Trans Patt Anal Mach Intell* 8(6): 679-698

---

# Hybrid Fuzzy Clustering Method

Tomasz Przybyła

Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland  
Tomasz.Przybyla@polsl.pl

**Summary.** A new hybrid clustering method based on a fuzzy myriad is presented. The proposed method could be considered as generalization of well known fuzzy  $c$ -means method (FCM) proposed by Bezdek. Existing modification of the FCM method, such as conditional clustering or partial supervised clustering can be applied to determine the objective function of the proposed method.

## 1 Introduction

The clustering aims at assigning a set of objects to clusters in such a way that objects within the same cluster have a high degree of similarity, while objects belonging to different clusters are dissimilar. The clustering methods can be divided into hierarchical and nonhierarchical (partitioning) methods. In this paper, clustering by minimization of a criterion function will be considered. The most traditional clustering methods are "hard" partitioning i.e. every object belongs to one group (cluster). In such a partition boundaries among clusters are sharp. However, in practice, the boundaries are not strict but ambiguous. Thus, soft partitioning is more suitable in this case. However, the fuzzy set theory proposed by Zadeh [3] performs soft partitioning. The most popular method of fuzzy clustering is the fuzzy  $c$ -means (FCM) method proposed by Bezdek [1]. Unfortunately the FCM method is sensitive to presence of outliers and noise in clustered data. In real applications, the data are corrupted by noise and assumed models such a Gaussian distribution are never exact. In this method is a prototype-based method, where prototypes are optimal for the Gaussian model of data distribution. The Gaussian model is inadequate in an impulsive environment. Impulsive signals are more accurately modeled by distributions which density functions have heavier tails than the Gaussian distribution [4, 5].

This paper is divided into four sections. In the section 2.1, the weighted myriad (treated as fuzzy myriad) and its properties are described. Next, in the section 2.2 is introduced an objective function and its optimal arguments: clusters prototypes and partition matrix. The subsection 2.3 describes the method of estimation of myriad linearity parameter value. The section 3 presents experimental results. Finally, in section 4 conclusions are presented.

## 2 Hybrid Fuzzy Clustering Method (HFCMyr)

### 2.1 Weighted Myriad

Let us consider a set of  $N$  independent and identically distributed (iid) observations,  $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ , and a set of assigned weights  $\mathcal{U} = \{u_1, u_2, \dots, u_N\}$ . A weighted myriad is the value, that minimizes the weighted myriad objective function defined as follows:

$$\hat{\Theta} = \arg \min_{\Theta \in \mathfrak{R}} \sum_{k=1}^N \ln \left[ K^2 + u_k (x_k - \Theta)^2 \right]. \quad (1)$$

The value of weighted myriad depends on the data set  $\mathcal{X}$ , the assigned weights  $\mathcal{U}$  and the parameter  $K$ , called a linearity parameter. Two interesting cases may occur. First, when the  $K$  value tends to infinity (i.e.  $K \rightarrow \infty$ ), then the value of weighted myriad converges with the weighted mean, that is

$$\lim_{K \rightarrow \infty} \hat{\Theta}_K = \frac{\sum_{k=1}^N u_k x_k}{\sum_{k=1}^N u_k}. \quad (2)$$

where  $\hat{\Theta}_K = \text{myriad} \{x_k \diamond u_k; K\}_{k=1}^N$ . This property is called myriad linear property [5].

Second case, called modal property, occurs when the value of  $K$  parameter tends to zero (i.e.  $K \rightarrow 0$ ). In this case the value of the weighted myriad is always equal to one of most frequent values in the input data set, i.e.:

$$\hat{\Theta}_0 = \arg \min_{x_j \in \Xi} \prod_{k=1, x_k \neq x_j}^N |x_k - x_j|. \quad (3)$$

where

$$\hat{\Theta}_0 = \lim_{K \rightarrow 0} \hat{\Theta}_K,$$

and  $\Xi$  is a set that contains the most frequent data in the input data set  $\mathcal{X}$ . The value  $\hat{\Theta}_K$  is defined in the same way as in the linear property.

### 2.2 Objective Function

Let's consider a clustering category in which partitions of data set are built on the basis of some performance index, known also as an objective function [2]. The minimization of a certain objective function can be considered as an optimisation approach leading to suboptimal configuration of the clusters. The main design challenge lies in formulating an objective function that is capable of reflecting the nature of the problem so that its minimization reveals a meaningful structure in the data set.

The proposed method is an objective functional method based on fuzzy c-partitions of finite data set [1]. The proposed objective function can be an extension of classical within-groups sum of squared error objective function.

Assuming that a different value of  $K$  has been assigned to each cluster, the objective function of proposed method can be described in the following way:

$$J_m(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^c \sum_{k=1}^N \sum_{l=1}^p \ln \left[ K_i^2 + u_{ik}^m (x_k(l) - v_i(l))^2 \right], \quad (4)$$

where  $c$  is the number of clusters,  $N$  is the number of data and  $p$  is the number of features.  $K_i^2$  is the myriad linear parameter assigned to the  $i$ -th cluster and  $1 \leq i \leq c$ ,  $u_{ik} \in \mathbf{U}$  is the fuzzy partition matrix,  $x_k(l)$  represents  $l$ -th feature of the  $k$ -th input data from the data set and  $1 \leq l \leq p$  and  $m$  -s a fuzzyfying exponent called fuzzyfier.

The optimization of the objective function  $J_m$  is completed with respect to the partition matrix and prototypes of the clusters. The first step is a constraint-based optimization, which involves Lagrange multipliers to accommodate the constraints of the membership grades [1, 2].

The columns of partition matrix  $\mathbf{U}$  are independent, so the minimization of objective function (4) can be described as:

$$\begin{aligned} J_m(\mathbf{U}, \mathbf{V}) &= \sum_{k=1}^N \sum_{i=1}^c \sum_{l=1}^p \ln \left[ K_i^2 + u_{ik}^m (x_k(l) - v_i(l))^2 \right] \\ &= \sum_{k=1}^N J_k \end{aligned} \quad (5)$$

The minimization of (4) can be reduced to minimization of  $N$  independent components  $J_k$ . When a linear transformation  $\mathcal{L}(\cdot)$  is applied to the expression  $K_i^2 + u_{ik}^m (x_k(l) - v_i(l))^2$  its variability range is changed to  $(0, 2]$ , i.e.

$$0 \leq \mathcal{L} \left( K_i^2 + u_{ik}^m (x_k(l) - v_i(l))^2 \right) \leq 2. \quad (6)$$

Based on (6) and representing the logarithm function by its power series, the minimized of the objective function can be reduced to the following expression:

$$J'_k = \sum_{i=1}^c \sum_{l=1}^p [L_i^2 + u_{ik}^m d_{ik}^2(l) - 1], \quad (7)$$

where

$$L_i^2 + u_{ik}^m d_{ik}^2(l) = \mathcal{L} \left( K_i^2 + u_{ik}^m (x_k(l) - v_i(l))^2 \right),$$

and

$$d_{ik}^2(l) = (x_k(l) - v_i(l))^2.$$

When the Lagrange multipliers optimization method is applied to the (7) expression we obtain:

$$J'_k(\lambda, \mathbf{u}_k) = \sum_{i=1}^c \sum_{l=1}^p [L_i^2 + u_{ik}^m d_{ik}^2(l) - 1] - \lambda \left( \sum_{i=1}^c u_{ik} - 1 \right), \quad (8)$$

where  $\lambda$  is the Lagrange multiplier,  $\mathbf{u}_k$  is the  $k$ -th column of partition matrix and the term  $\lambda(\sum_{i=1}^c u_{ik} - 1)$  arises from definition of the partition matrix  $\mathbf{U}$  [1, 2]. When the gradient of (8) is equal to zero then, for the sets defined as:

$$\begin{aligned} \forall_{1 \leq k \leq N} \mathcal{I}_k &= \{i \mid 1 \leq i \leq c; \|\mathbf{x}_k - \mathbf{v}_i\|^2 = 0\} \\ \tilde{\mathcal{I}} &= \{1, 2, \dots, c\} - \mathcal{I}_k, \end{aligned}$$

the values of partition matrix are described by

$$\forall_{1 \leq i \leq c} \forall_{1 \leq k \leq N} u_{ik} = \begin{cases} \left[ \sum_{j=1}^c \left( \frac{\|\mathbf{x}_k - \mathbf{v}_i\|}{\|\mathbf{x}_k - \mathbf{v}_j\|} \right)^{2/(m-1)} \right]^{-1} & \text{if } \mathcal{I}_k = \emptyset \\ 0 & \text{if } \forall_{i \in \tilde{\mathcal{I}}_k} \\ 1 & \text{if } \mathcal{I} \neq \emptyset. \end{cases} \quad (9)$$

where  $\|\cdot\|$  is an Euclidean norm, and  $\mathbf{v}_i$  are cluster prototypes  $1 \leq i \leq c$ .

For the fixed number of clusters  $c$ , parameters  $m$  and  $K_i$   $1 \leq i \leq c$  and the fixed partition matrix  $\mathbf{U}$ , the prototype values minimizing (4) are fuzzy myriads described as follows:

$$v_i(l) = \arg \min_{\Theta \in \mathfrak{R}} \sum_{k=1}^N \ln \left[ K_i^2 + u_{ik}^m (x_k(l) - \Theta)^2 \right], \quad (10)$$

where:  $i$  is the cluster number  $1 \leq i \leq c$  and  $l$  is the component (feature) number  $1 \leq l \leq p$ .

### 2.3 K Value Estimation

The  $\alpha$ -stable distribution is a generalization of the Gaussian distribution (when  $\alpha = 2$ ) or the Cauchy distribution (when  $\alpha = 1$ ). So, methods for evaluating parameters of  $\alpha$ -stable distribution can be applied for the mentioned distributions.

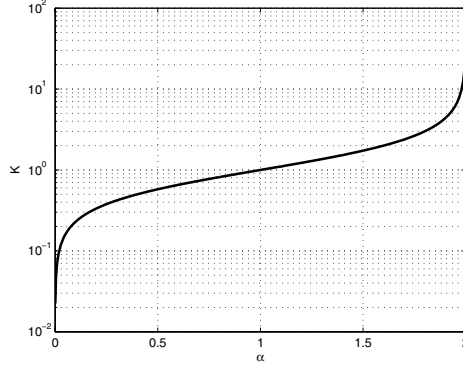
Assuming, that  $\mathbf{x}$  is  $\alpha$ -stable random variable, and  $\mathbf{y} = \ln |\mathbf{x}|$ , the following dependence can be proofed [6]

$$Var(\mathbf{y}) = \frac{\pi^2}{6} \left( \frac{1}{\alpha^2} + \frac{1}{2} \right), \quad (11)$$

where  $0 \leq \alpha \leq 2$ . For the Gaussian distribution ( $\alpha = 2$ ), the  $K$  value should approach infinity. For  $K > 50$ , differences between the fuzzy myriad and the fuzzy mean can be omitted, therefore at the Gaussian point is set a  $K = 50$ . For the  $\alpha < 1$ , the fuzzy myriad estimator should be as selective as possible ( $K$  should approach 0). The following relation between  $\alpha$  and  $K$  values has been proposed [7]

$$K = \sqrt{\frac{\alpha}{2 - \alpha}}. \quad (12)$$





**Fig. 1.** Proposed  $\alpha - K$  curve

For the data set  $\mathcal{X}$ , the estimation for each cluster can be done in the following way

- For the  $i$ -th cluster, for each element  $1 \leq l \leq p$  of feature vectors belonging to the  $i$ -th cluster, compute the  $K_l$  parameter,
- Finally, the value of  $K_i$  is computed as  $K_i \min_{1 \leq l \leq p} K_l$ .

The proposed strategy of the  $K_i$  value estimation can be considered as a maximum selectivity strategy.

In some cases, an estimation of the  $\alpha$  value based on (11) results incorrect values (i.e. the  $\alpha$  is the complex value). In such cases, the value of  $K_i$  parameter is fixed at 2.0.

## 2.4 Clustering Data with HFCMyr Clustering Method

The hybrid clustering method can be described as follows:

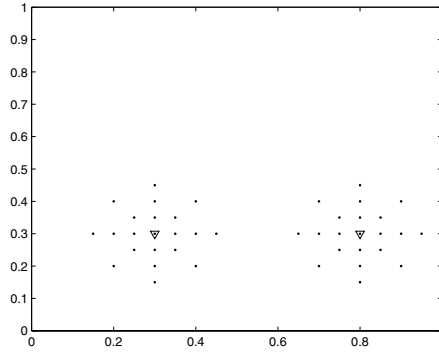
1. For the given data set  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  where  $\mathbf{x}_i \in \mathbb{R}^p$ , fix the number of clusters  $c \in \{2, \dots, N\}$ , the fuzzyfing exponent  $m \in [1, \infty)$  and the tolerance limit  $\varepsilon$ . Initialize randomly the partition matrix  $\mathbf{U}$  and fix initial values of  $K_i$  parameter for each cluster, fix  $l = 0$ ,
2. calculate the prototype values  $\mathbf{V}$ , as weighted myriads. A weighted myriad has to be calculated for each feature of  $\mathbf{v}_i$  based on (10),
3. update the partition matrix  $\mathbf{U}$  using (9),
4. update  $K_i$ , where  $1 \leq i \leq c$  based on (11) and (12),
5. if  $\|\mathbf{U}^{(l+1)} - \mathbf{U}^{(l)}\| < \varepsilon$  then stop the clustering algorithm, otherwise  $l = l + 1$  and go to (2).

## 3 Numerical Experiments

To illustrate the performance of proposed method, HFCMyr method is run on two data sets. The first data set is an artificial two-group data in  $2D$  space. The

well known IRIS data set proposed by Fisher [8] has been chosen as the second data set.

In all evaluated simulations the fuzzyfier  $m = 2$  was used. The Fuzzy C-Means method, proposed by Bezdek [1], has been chosen as the reference method. The figure 2 shows the first data set. This data set is fully synthetic with well-defined centers of groups in  $2D$ , located at  $[0.3 \ 0.3]^T$  and  $[0.8 \ 0.3]^T$  respectively. The true value group centers are marked with triangles. Outliers have been added at the point  $[100, 100]^T$ . The number of outliers varies from 0 (no outliers) up to 32 (number of points when the proposed method has crashed down). The test data set with outliers has not been presented because of its illegibility. For the synthetic data set, the following values have been fixed: the number of clusters  $c = 2$ , and the tolerance limit  $\varepsilon = 10^{-6}$ . The results for the proposed method are presented in the table 1, and the results of reference method in the table 2. For the IRIS data set, the following values have been fixed: the number of clusters  $c = 3$ , and the tolerance limit  $\varepsilon = 10^{-6}$ .



**Fig. 2.** The synthetic data set

**Table 1.** The prototype values of synthetic data set and the proposed method

Number of outliers	HFCMyr			
	$\mathbf{V}_1$	$K_1$	$\mathbf{V}_2$	$K_2$
0	$[0.2987 \ 0.3000]^T$	2.0	$[0.8013 \ 0.3000]^T$	2.0
5	$[0.3103 \ 0.3113]^T$	2.0	$[0.8028 \ 0.3011]^T$	0.6454
10	$[0.3219 \ 0.3227]^T$	2.0	$[0.8036 \ 0.3016]^T$	0.5672
15	$[0.3336 \ 0.3342]^T$	2.0	$[0.8044 \ 0.3022]^T$	0.5469
20	$[0.3455 \ 0.3459]^T$	2.0	$[0.8050 \ 0.3029]^T$	0.5423
25	$[0.3578 \ 0.3582]^T$	2.0	$[0.8056 \ 0.3036]^T$	0.5443
30	$[0.3708 \ 0.3711]^T$	2.0	$[0.8058 \ 0.3043]^T$	0.5495
32	$[0.5500 \ 0.3000]^T$	2.0	$[100.0 \ 100.0]^T$	2.0

**Table 2.** The prototype values for synthetic data set and the reference method

Number of outliers	$\mathbf{V}_1$	$\mathbf{V}_2$
0	$[0.3000 \ 0.3000]^T$	$[0.8000 \ 0.3000]^T$
5	$[0.5500 \ 0.3000]^T$	$[100.0 \ 100.0]^T$

Results are shown in the tables 3 and 4 for the proposed and the reference methods, respectively.

**Table 3.** Results for IRIS data set and the proposed method

	I	II	III	$K$
$\mathcal{X}_1$	50			2.0
$\mathcal{X}_2$	45	5		2.0
$\mathcal{X}_3$	7	43		2.0

**Table 4.** Results for IRIS data set and the reference method

	I	II	III
$\mathcal{X}_1$	50		
$\mathcal{X}_2$	47	3	
$\mathcal{X}_3$	13	47	

The results obtained from proposed method and the reference method are very similar for the data without outliers. An increase the number of outliers improve selectivity of clusters estimation in the proposed method (the value of  $K_i$  parameter decreases), while in the reference method incorrect (different than expected) results are obtained. The critical number of outliers that causes breaking results from the proposed method is 31, whereas the reference method breaks down for data with 5 outliers.

## 4 Conclusions

This paper has dealt with clustering of data corrupted by noise and outliers. The well-known existing methods, for example Bezdek's FCM are sensitive on outliers, hence the obtained groups can be different than primary expected. Therefore, methods with intuitively correct results (the same or very similar to expected) are worth searching for.

Results of the proposed method are more accurate than the reference method outputs. A nonlinear estimation of group prototypes has increased robustness of the clustering method. The main advantage of proposed method is its flexibility, i.e. for each cluster the same estimation method is applied for the prototype

value assessment. From the estimation point of view, the values of prototypes are computed in different manners. For small values of  $K$  parameter (i.e. below 2.0) the prototype value is computed as nonlinear, very selective estimator. For large values of  $K$  parameter (i.e. greater or equal 50) the prototype values are computed as ordinary mean. The cost of the increased robustness and flexibility is the longer computation time.

## Acknowledgment

The author would like to thank the anonymous reviewers for their valuable comments.

## References

1. Bezdek J.C. (1981) Pattern Recognition With Fuzzy Objective Function Algorithms, Plenum Press, New York
2. Pedrycz W. (2005) Knowledge-Based Clustering, Wiley-Interscience
3. Zadeh L.A. (1965) Fuzzy Sets, Information and Control 8:338-353
4. Arce G.R., Kalluri S. (2000) Fast Algorithm For Weighted Myriad Computation by Fixed Point Search, IEEE Trans. on Signal Proc. 48:159-171
5. Arce G.R., Kalluri S. (2001) Robust Frequency-Selective Filtering Using Weighted Myriad Filters admitting Real-Valued Weights, IEEE Trans. on Signal Proc. 49:2721-2733
6. Georgiu P.G., Tsakalides P., Kyriakakis C. (1999) Alpha-Stable Modeling of Noise and Robust Time-Delay Estimation in the Presence of Impulsive Noise, IEEE Trans. on Multimedia 1:291-301
7. Gonzales J.G., Arce G.R. (2001) Optimality of the Myriad Filter in Practical Impulsive-Noise Environments. IEEE Trans. on Signal Proc. 49:438-441
8. Fisher R.A. Iris Data Set, <http://www.ics.uci.edu/~mllearn/MLRepository.html>

---

# Structural Image Analysis Based on Ontological Models

Juliusz L. Kulikowski

Institute of Biocybernetics and Biomedical Engineering Polish Academy of Sciences, Ks. Trojdena Str.4 02-109 Warsaw, Poland  
jlkulik@ibib.waw.pl

**Summary.** A concept of structural image analysis and interpretation based on super-relations and on ontological models is presented. The system of image interpretation should contain the structural analysis (SA), ontological models (OM) and semantic relationships (SR) modules. The role of modules is described. The proposed approach is illustrated by an example of cardiac USG images interpretation.

## 1 Introduction

The idea of structural approach to pattern recognition arose in early 60ths years of the past century, when it became evident that “standard” pattern recognition methods based on geometrical, statistical, potential functions etc. approaches as well as those based on artificial neural networks are ineffective in composite form patterns recognition. First attempts to solve the problem were based on formal linguistic concepts considering patterns as expressions of a language defined by a formal grammar [1][2] [3][4]. Examples of formal linguistic methods application to structural image analysis with comments concerning some drawbacks and possible extensions of linguistic methods have been presented by K. S. Fu (1974, [5]). Substantial contributions to linguistic methods in image analysis and their applications have also been done in [6][7][8][9]. All the above-mentioned works concerned structural analysis of binary (black-white) images. An alternative approach to structural image analysis, based on an extended algebra of relations, was proposed in [10][11]. Unlike linguistic methods this approach admitted structural description of multilevel images without their binarization. The extension of the algebra of relations admitted algebraic operations performed on relations described on different families of sets. This approach then has been extended on any non-homogenous spaces of image features [12] and on hyper-relations [13]. However, a common drawback of both formal linguistic and algebraic methods of structural analysis of images consists in difficulty of effective including context knowledge into the image analysis algorithms. The problem is unnoticeable in simple pattern recognition, when the number of discriminated classes of objects is strongly limited and the meaning of the classes have been a priori defined.

However, it becomes substantial when image understanding problem is posed and answering a question: “What situation is presented by the image?” rather than: “What is it?” is required [14] [15]. One of possible ways to include context knowledge into image analysis procedures consists in combining structural image analysis algorithms with ontologies of the given area of interest (a part of real world under consideration). This concept can be effectively realized when ontology of a given application domain is presented in the form of a structure of relations-based ontological models [16].

The aim of this paper is presentation the above-mentioned concept in a more detailed form. In Sec. 2 a general concept of image understanding system based on ontological models is presented. In Sec. 3 the concept is exemplified by an ultrasound cardiac images interpretation system. Conclusions are formulated in Sec. 4.

## 2 Image Understanding Supported by Ontological Models

**Image interpreting system.** We consider image interpreting system as a system able to give logically justified answers on a class  $Q$  of queries concerning a set  $V$  of images provided from an application area. The application area will be denoted by  $A$  and it will be assumed that it is a part of a real or abstract world about which certain knowledge is a priori given in the below presented form.

The class  $Q$  of queries may concern, in particular [17]:

- image features (geometrical, morphological, spectral, statistical, etc.);
- image meaning or content;
- relations of a given image to other images;
- image quality and/or utility for a given set of users.

In this paper attention will be focused on the second subclass of queries i.e. on the relationships between images and represented by them objects belonging to a given application area  $A$ . Image *understanding* will be thus considered as ability to answer this type of queries. A philosophical problem whether a computer system understanding image in the above-defined sense is *conscious* of the image content or *self-conscious* of its ability to understand the image, is here beyond our area of interest; our aim consists in design of computer systems aiding image analysis rather than in removing humans from this process. The problems of image acquisition, enhancement and preliminary processing (including features extraction) are here also out of interest.

A logical core of a computer-aided image understanding system should contain the modules:

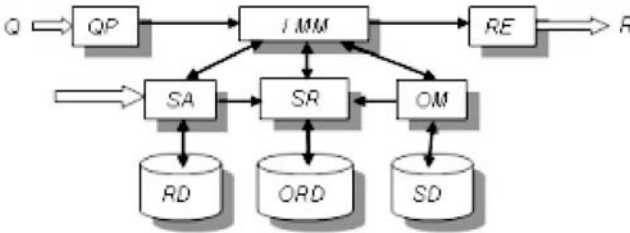
- $SA$  – of structural analysis of the class  $V$  of images;
- $OM$  – of ontological models describing the application area  $A$ ;
- $SR$  – of semantic relationships between objects derived from  $V$  and  $A$ .

The modules should be completed, correspondingly, by databases:

- $RDB$  – of reference images;

*ORDB* – of ontological relations;  
*SDB* – of semantic relations

The system should be managed by a logical managing module *LMM* connected on one side with a parser of queries *QP* and with an editor of replies *RE* on the other side. Images *I* and queries *Q* are input data and replies *R* are output data of the system, as shown in Fig. 1.



**Fig. 1.** A scheme of computer-aided image understanding system

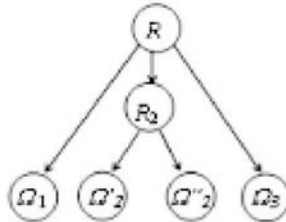
In traditional computer-aided systems of image analysis the modules *SR* and *OM* are reduced to a minimum. Such systems are able not more but to rendering enhanced image, evaluated image parameters or names and position of recognized objects. Advanced image interpretation needs combining image processing procedures with the *OM* module representing in a structured form the application area knowledge. The *SR* module plays the role of a logical interface between the *SA* and *OM* modules. We would like to show that formal structure of the *SA*, *OM* and *SR* modules can be based on the relations and super-relations concepts. Before a more detailed description of this idea several basic notions should be reminded.

**Relations and super-relations.** If  $\Omega_1, \Omega_2, \dots, \Omega_n$  are some sets belonging to  $F$  then a *relation* described on the (linearly ordered) family of sets  $[\Omega_1, \Omega_2, \dots, \Omega_n]$  is defined as any subset of their Cartesian product:

$$r \subseteq \Omega_1 \times \Omega_2 \times \dots \times \Omega_n \tag{1}$$

and any  $n$ -tuple  $[\xi_1, \xi_2, \dots, \xi_n] \in r$  is called a *syndrome* of  $r$ . The natural number  $n$  of sets in expression (1) will be called *order* of the Cartesian product and of the relation. From a formal point of view any set  $\Omega_i$  is a first-order Cartesian product and any set can be considered as a first-order relation as well. Such sets, considered as first-order Cartesian products, will be called *ordinary* sets. Due to the fact that relations have been defined as sets, some of the above mentioned sets  $\Omega_1, \Omega_2, \dots, \Omega_n$  may be relations themselves. This leads to the concept of relations between relations or of *super-relations* (*s-relations*). The syndromes of super-relations take the form of linearly ordered bracket-structures; e.g. an expression  $[\xi_1, \rho_2, \xi_3]$  may represent a syndrome of a relation  $R$  described on a Cartesian product  $\Omega_1 \times R_2 \times \Omega_3$  where  $R_2 \subseteq \Omega'_2 \times \Omega''_2$  has been defined itself

as a relation. In such case the relation  $R_2$  is called to be *nested* in the given higher-level relation  $R$ . Multi-level nesting of super-relations is also admissible. A relation described on a Cartesian product of ordinary sets only will be called a *1<sup>st</sup> level* Cartesian product and any relation described on a *1<sup>st</sup> level* Cartesian product will be called a *1<sup>st</sup> level s-relation*. A relation defined on a Cartesian product of ordinary sets and/or of any number of *1<sup>st</sup> level* only super-relations will be called a *2<sup>nd</sup> level s-relation* (e.g. the *s-relation* whose syndrome has been mentioned above). In general, a relation described on a Cartesian product of sets and of nested in it at most  $(k-1)^{th}$  level *s-relations* will be called a *k<sup>th</sup> level s-relation*. The inner structure of a *k<sup>th</sup> level s-relation* can be represented by a directed graph. In simple cases the graph takes the form of a rooted tree whose root corresponds to the *s-relation* as a whole and nodes correspond to the ordinary sets and nested *s-relations*, the leaves being assigned to the ordinary sets only. The edges connect nodes assigned to the top- and the nested *s-relations* with those assigned to the elements of the corresponding Cartesian products. A tree representing the above-mentioned *2<sup>nd</sup> level s-relation*  $R$  is shown in Fig. 2.



**Fig. 2.** Structure of a *2<sup>nd</sup> level s-relation*

In more sophisticated cases trees are unable to represent the inner structure of *s-relations*. Such situation is illustrated in Fig. 3. Here, a *3<sup>rd</sup> level s-relation*  $R$  has been defined as an algebraic composition:

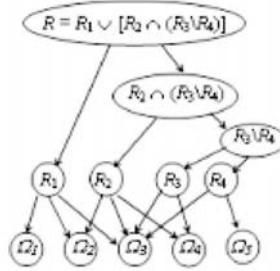
$$R = R_1 \cup [R_2 \cap (R_3 \setminus R_4)]$$

where  $R_1$  is a first-level relation described on ordinary sets  $\Omega_1, \Omega_2,$  and  $\Omega_3,$   $R_2$  is a *s-relation* described on ordinary sets  $\Omega_2, \Omega_3$  and  $\Omega_4,$   $R_3$  is a relation described on ordinary sets  $\Omega_3$  and  $\Omega_4,$   $R_4$  is a relation described on ordinary sets  $\Omega_3$  and  $\Omega_5.$  All algebraic operations are considered in the extended relations algebra sense [11][12].

Another situation arises when several are defined on the same family of sets or lower-level relations; in such case the *s-relations* are represented by a sum of the corresponding graphs.

Therefore, if there are given: a family  $F$  of ordinary sets and a family  $\Psi$  of relations or *s-relations* defined on any sub-families of  $F$  then it is possible to construct a directed graph  $\Gamma$  representing the structure as well as formal connections between the elements of  $\Psi.$





**Fig. 3.** A graph of an 4-th order *s*-relation described by an algebraic combination of lower-level relations

This, finally, leads to a general formal concept of the *SA*, *SR* and *OM* modules as triples:

$$M^* = [F^*, \Psi^*, \Gamma^*] \tag{2}$$

where  $F^*$  denotes a family of sets of basic objects,  $\Psi^*$  - a set of relations and/or *s*-relations described on selected sets of  $F^*$ ,  $\Gamma^*$  - a graph describing logical relationships between the elements of  $\Psi^*$ . The star \* stands here for *SA*, *SR* or *OM*, respectively.

**Indeterminate relations.** The deterministic formalism of relations and/or *s*-relations can be extended on indeterminate relations (*s*-relations). For this purpose to the family  $F$  a set  $V$  of real numbers,  $V = [0, \dots, 1]$  should be included. Then, instead of relations defined by (1) the following ones:

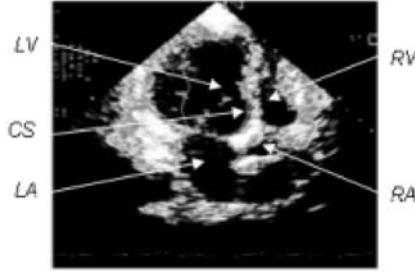
$$w \subseteq \Omega_1 \times \Omega_2 \times \dots \times \Omega_n \times V \tag{3}$$

can be taken into consideration. Their syndromes will thus take the form  $[\xi_1, \xi_2, \dots, \xi_n; v]$ , where  $v, v \in V$ , is a measure of the relation being satisfied by the given syndrome. This approach is similar to this of fuzzy sets of L. A. Zadeh [19]. However, in this case some additional constraints imposed on the measures  $v$  follow from the extended algebra of relations. In particular, if  $w'$  and  $w''$  are two indeterminate relations and the corresponding syndromes  $[\xi_1, \xi_2, \dots, \xi_n; v']$ , and  $[\xi_1, \xi_2, \dots, \xi_n; v'']$  are considered then for a third relation  $w$  and its syndrome  $[\xi_1, \xi_2, \dots, \xi_n; v]$  it should be:

$$\begin{aligned} \text{if } w &= \neg w' && \text{then } v = 1 - v', \\ \text{if } w &= w' \cup w'' && \text{then } v = \max(v', v''), \\ \text{if } w &= w' \cap w'' && \text{then } v = \min(v', v'') \end{aligned} \tag{4}$$

### 3 Application to Medical Image Analysis

Let us assume that  $V$  is a class of ultrasound cardiac images used in medical diagnosis. A typical image of this type is shown in Fig. 4.



**Fig. 4.** USG cardiac image: *LV* – left ventricle, *LA* – left atrium, *RV* – right ventricle, *RA* – right atrium, *CS* – cardiac septum. Visible automatically plotted contour of *LV*.

The problem consists, in general, in automatic recognition, identification and measurement of morphological parameters and diagnostic evaluation of basic anatomic details: ventricles (*LV*, *RV*), atria (*LA*, *RA*), cardiac septum (*CS*), etc. In human-computer interaction based diagnostic systems a great deal of knowledge concerning the observation area is introduced in the form of auxiliary human control decisions (see, for example [18]). In fully automatic diagnostic systems a similar type of information should be introduced in the form of data and of parameters specifying the decision algorithms. Our attention will be focused below on automatic identification, contouring and measuring of geometrical parameters of left heart ventricle (*LV*) in the USG images, as a preliminary step to calculation of some hemodynamical parameters for medical diagnosis.

The *SA* module of the system should contain a set of *s*-relation-based procedures of the following types:

- recognizing, delimiting and/or localizing: *background*, *object on the background*, *hole(s) in the object*, *walls surrounding a hole*, etc.;
- measuring parameters characterizing *size* and *form* of the above-mentioned objects;
- establishing geometrical relationships between the recognized objects: *A larger than B*, *A smaller than B*, *A close to B*, *A far to B*, etc.;
- establishing topological relationships between the recognized objects: *A inside B*, *A adjacent to B*, *A above B*, *A below B*, *A on the left side of B*, *A on the right side of B*, etc..

On the other hand, the *OM* module should represent basic medical knowledge about human heart, in particular its *anatomy*, *physiology*, *functions*, *diseases*, *methods of examination*, *cardiac visualization modalities and methods*, etc. Heart anatomy can be represented in the form of a *mereological taxonomy*:

*Human heart:*

- *myocardium*
- *cavities*
- *valves*
- *arteries*

which can be extended, say,

- *heart cavities: LV, LA, RV, LA*

The geometrical and topological relationships in the *SA* module can also be included into the *OM* module and applied to the anatomical objects.

At last, the *SR* module should contain relations between *geometrical objects* recognized in the *SA* module and *anatomical objects* described by the *OM* module. Such relations should be established, for example between:

- *object on the background and human heart,*
- *hole(s) in the object and cavities,*
- *walls surrounding a hole and myocardium  $\cup$  valves,*

etc. Then any information (element of medical knowledge) concerning e.g. the *left heart ventricle (LV)* given in the form of *s-relations* in which the *LV* term is involved can be connected with an object *U* recognized in the *SA* module due to a chain of intersections of relations:

$$[T, \text{object}, X] \cap [X, \text{hole in}, T] \cap [X, \text{on the left side}] \cap [X, \text{on the upper side}] \cap [X, \text{on the left side, on the upper side, LV}] \cap [LV, \text{getting blood from, LA}] \cap [LV, \text{entering blood to, AO (aorta)}],$$

etc. This leads to the conclusion that the recognized object *X participates in providing blood from LA to AO.*

## 4 Conclusions

It has been shown in the paper how the formalism of *s-relations* combined with ontological models can be used to effective image interpretation. Further investigations of the proposed approach are needed.

## References

1. Ledley R.S., Rotolo L.S., Golab T.J. et al (1965) FIDAC: film input to digital automatic computer and associated syntax-directed pattern recognition programming system. In "Optical and Electro-Optical Information Processing" (Tippet et al eds.), Chapt. 33, MIT Press, Cambridge Mass., .591-614.
2. Shaw A.C. (1968) The formal description and parsing of pictures. Rep. SLAC-84, Stanford Linear Accelerator Center, Stanford Univ., Stanford Calif.
3. Pfaltz J.L., Rosenfeld A. (1969) WEB grammars. Proc. Int. Joint Conf. Artificial Intelligence 1<sup>st</sup>. Washington D.C., 609-619.
4. Feder J. (1971) Plex languages. Information Sci. vol. 3, 225-241.
5. Fu K. S. (1974) Syntactic methods in pattern recognition. Academic Press, New York.
6. Flasiński M. (1988) Parsing of edNLC – graph grammar for scene analysis. Patt. Recognition, vol. 21. nr 6.

7. Jakubowski R. (1985) Extraction of shape features for syntactic recognition of mechanical parts. *IEEE Trans. Syst. Man Cybern.*, SMC-15, vol. 5.
8. Grabska E., Hliniak G. (1992) Design space and graph grammars. *Machine Graphics and Vision*, vol. 1 nos 1-2, 25-34.
9. Ogiela M.R., Tadeusiewicz R., Ogiela L. (2005) Intelligent semantic information retrieval. In *Medical Pattern Cognitive Analysis. Computational Science and its Applications – ICCSA*, 852-857.
10. Kulikowski J.L. (1970) Some problems of structural analysis of composite patterns (in Polish). *Arch. Automatyki i Telemechaniki*, vol. XV, no 3.
11. Kulikowski J. (1971) Algebraic methods in pattern recognition. *CISM, Udine, Courses and Lectures No 85*. Springer Verlag, Wien.
12. Kulikowski J.L. (2002) Relational approach to structural analysis of images. *Machine Graphics and Vision*, vol.1, nos 1-2 , 299-309
13. Kulikowski J.L. (2004) Description of irregular composite objects by hyper-relations. In *Computer Vision and Graphics. Int. Conf. ICCVG 2004, Warsaw* (K. Wojciechowski et al eds.). Springer, 141-146.
14. Kulikowski J.L. (2002) From pattern recognition to image interpretation. *Biocybernetic and Biomedical Engineering* vol. 22, nos 2-3, 177-197.
15. Tadeusiewicz R., Ogiela M.R. (2004) *Medical image understanding technology*. Springer Verlag, Berlin.
16. Kulikowski J.L. (2005) The Role of Ontological Models in Pattern Recognition. In *Computer Recognition Systems. Proc. of the 4th International Conference on Computer Recognition Systems CORES'05* (M. Kurzynski et al eds), Springer; 43-52.
17. Kulikowski J.L. (2007) Image Understanding as a Step to Image Utility Assessment. *IEEE International Workshop on Imaging Systems and Techniques – IST 2007, Krakow, May 4-5, 2000*. CD: 1-4244-0965-9/©2007 IEEE.
18. Kulikowski J.L., Przytulska M., Wierzbicka D. (2001) Investigation of irregular shapes's time-variations in the left cardiac ventricle evolution. *IFMBE Proc., IX Mediterranean Conf. on Medical and Biological Eng. and Computing, MEDICON, Pula, Croatia, 12-15 June 2001, Part I*, 531-533.
19. Zadeh L.A.(1965) Fuzzy sets. *Information and Control*, vol. 8, no 3, 338-353.

---

# Homogeneity of Pixel Neighborhoods in Gray Level Images Investigated by the Grade Correspondence Analysis

Maria Grzegorek

Institute of Computer Science, Polish Academy of Sciences, Ordona 21, 01-237  
Warsaw  
mary@ipipan.waw.pl

**Summary.** The paper presents some interesting properties of Grade Correspondence Analysis (GCA) used to order the pixels of gray level images described by their gray level ( $gl$ ), gradient module ( $gm$ ) and a family of variables dealing with pixel neighborhoods. The  $gm$  stripcharts for pixels ordered according to GCA are used to separate and then order subsets of pixels with identical descriptions of their  $gm$  neighborhoods. The spatial structure of clusters formed from subsets of adjacent clusters is then restored and used to discover compact areas of pixels with similar  $gm$  neighborhoods. Such a decomposition is performed for a MR image of a fragment of human brain. Two similar homogeneous compact areas discovered in this image suggest that the proposed method might be useful in medical classification and identification.

## 1 Introduction

Grade Correspondence Analysis (GCA) is of special importance among data analysis methods known as grade- or copula-based (i.e. based on various transformations of datasets onto copulas). Grade models and methods were comprehensively described in [1]; the GradeStat program ([2]) implements these methods and can be downloaded as free software. Some examples of grade data exploration in medical and demographic domains are described in [3] and [4]; multivariate data originated from images and processed involving GCA were explored in [5], [6] and [7]. The variables used to describe images, proposed in [5], are recalled in Section 2. They are: gray level ( $gl$ ), gradient module ( $gm$ ) and a vector of seven variables dealing with pixel neighborhoods. The reordering of rows and columns due to GCA is recalled at the beginning of Section 3. Stress is laid on similarity of the  $gm$  stripcharts for various gray level images, in particular on the appearance of characteristic "quills" visible in Figures 1 and 3. Up to now, the nature of quills was not fully recognized although an attempt to describe them by the values of the neighborhood variables was made in [7]. Section 3 reveals that each quill consists of pixels with the  $gm$  values which have the same record of the neighborhood variables. Moreover, the quills are stochastically ordered as explained in Section 3. It is said that the only differences between the  $gm$  stripcharts for various images concern numbers and fractions of pixels in particular quills, their shapes and the maximal values of  $gm$  in the image.

Section 4 presents image decomposition due to *clusters of quills with adjacent records of neighborhood variables*. An image is decomposed into subimages which restore spatial structure of particular clusters. This is exemplified by three subimages obtained from a MR image of a fragment of human brain. The results are compared and discussed; special attention is directed towards compact areas with homogeneous *gm* neighborhoods.

Closing Section 5 suggests possible changes in the neighborhood variables, in the size of pixels neighborhoods and in cluster formation.

## 2 From a Gray Level Image to a Dataset with Information on Gradient Modules in Pixel Neighborhoods

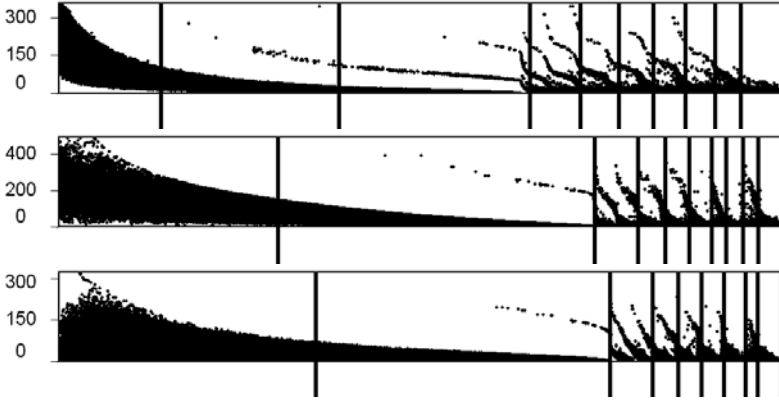
As mentioned in Section 2, the whole set of variables contains the pixel's gray level *gl*, the gradient module *gm* which reflects changes in gray level in the neighborhood of the pixel (obtained due to the Sobel operator [8]), and the neighborhood variables  $n_1, \dots, n_7$  constructed for a chosen value of parameter *b* being a promile of the maximal value of the gradient module denoted  $gm_{max}$ . The vector of thresholds  $t_i = i \times b \times gm_{max}, i = 1, \dots, 7$  serves to compare pixel's *gm* with the *gm* values of its neighbors in the image. Namely, the distance between the value of pixel's *gm* and the *gm* of each of the neighboring pixels is compared with the threshold and it is found out how many neighboring pixels have their *gm* distance less than the value of this threshold. The neighborhood variable  $n_i (i = 1, \dots, 7)$  is defined as the number of neighboring pixels with the *gm* distance smaller than the value of threshold  $t_i$ . Thus a pixel with  $(n_1, \dots, n_7) = (8, 8, 8, 8, 8, 8, 8)$  has all pixels in its neighborhood with the *gm* distance smaller than  $t_1$ ; a pixel with record  $(7, 8, 8, 8, 8, 8, 8)$  has seven neighboring pixels with *gm* distance smaller than  $t_1$  and with one pixel between  $t_1$  and  $t_2$ , and so on; finally, a pixel with record  $(0, 0, 0, 0, 0, 0, 0)$  has all neighboring pixels with the *gm* distance exceeding  $t_7$ .

In Sections 3 and 4 *b* is chosen to be 0.2.

## 3 The *gm* and *gl* Stripcharts for Pixels Ordered by GCA

Columns (variables) in the dataset described in Section 2 are primarily normalized: each cell is divided by the total of the column to which the cell belongs and then also by the total of the respective row. Such matrix is called the *overrepresentation matrix*; its chart (overrepresentation map) is a powerful tool in datasets exploration ([1][2]). The GCA procedure reorders rows and columns of such matrix to obtain a maximal value of a chosen positive dependence measure (usually Spearman or Kendall correlation). This ensures that adjacent rows (pixels) tend to be similar one to another. Procedure GCA is then completed by post GCA clustering which groups adjacent pixels into an arbitrarily chosen number of disjoint subsets visualized as separate subimages ([4], [5]).

Stripcharts of variables  $gm$  and  $gl$  can be effectively used to present the dataset reordered by GCA. Figure 1 shows the  $gm$  stripcharts for three images: the well known image "Lena", the image of a crossroad and the MR image of a knee.

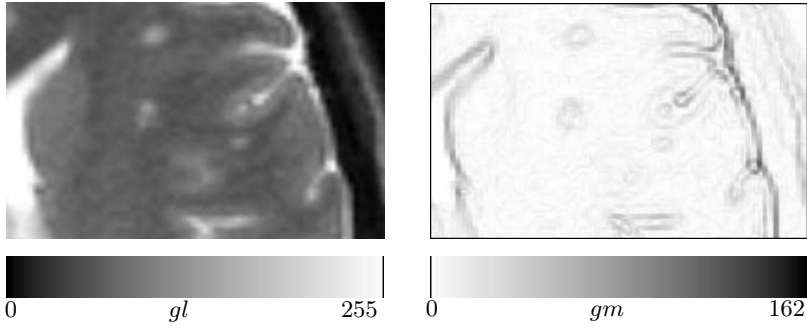


**Fig. 1.** The  $gm$  stripcharts for pixels ordered by GCA in three images: "Lena", "a crossroad", "a knee" (10 clusters formed in each image for adjacent pixels)

Points on the horizontal axes in Figure 1 represent pixels ordered according to GCA when  $n_1, \dots, n_7$  are constructed with  $b = 0.2$ . Vertical lines separate clusters, 10 in each example but consisting of different numbers (and also different fractions) of pixels. Although the values of  $gm_{max}$  are different for each image (354.5, 492.7, 323.1) and so are the total numbers of pixels, the shapes of the  $gm$  stripcharts are similar (as well as in other images investigated for  $b = 0.2$ ). This remarkable fact will be now explained in the case of a fragment of a human brain MR image, presented in Figure 2 at left.

The graph of variable  $gm$  for pixels in this image is shown in Figure 2 at right. The maximal value of  $gm$  is 161.1, but only a small fraction consisting of 465 pixels (out of 61936) attains the  $gm$  values exceeding 100 which makes them well visible at the  $gm$  graph.

The largest  $gm$  values form several contour curves corresponding to the borders of the areas with highest values of  $gm$ . The areas with similar  $gl$  values are visible in Figure 2 at left. The pixels in the image are located in the rectangle containing points  $(x, y)$  such that  $0 \leq x \leq 320$  and  $0 \leq y \leq 200$ . Our attention will be concentrated on two small sets of pixels, placed one under another in the upper part of the image. They are close to circles with centers at  $(x, y) = (126, 174)$  for the upper set and  $(x, y) = (120, 110)$  for the lower set. The diameters of the  $gl$  circles are equal to about 16, and the values of  $gl$  are in both circles close to 145 as follows from the scale placed in Figure 2 under the left rectangle. The respective  $gm$  circles have the  $gm$  values around 20; they are



**Fig. 2.** Fragment of a human brain on MR image; left: gray level ( $gl$ ), right: gradient module ( $gm$ )

surrounded by the narrow rings of pixels with larger  $gm$  values, well visible at the  $gm$  graph in Figure 2 right.

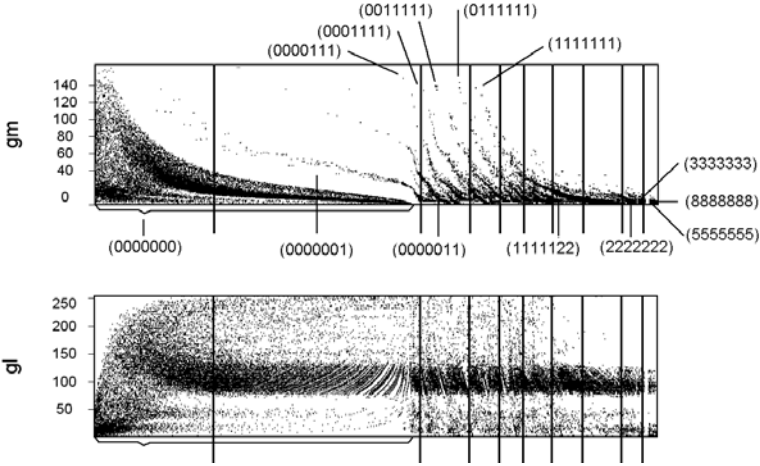
Both circles were indicated by a medical consultant as interesting from the medical point of view. In Section 4 they will be also shown very interesting from the data analyst's point of view as areas of pixels with  $gm$  neighborhoods differentiated similarly (and rather strongly).

The  $gm$  and  $gl$  post GCA stripcharts obtained from the image presented in Figure 2 are shown in Figure 3. The vertical lines indicate the ends of each of ten clusters formed from pixels which are adjacent according to GCA. The shape of the  $gm$  stripchart is similar to the top  $gm$  stripchart in Figure 1 (corresponding to "Lena"), although the two images (a photo and a MR image) are really very different. In both  $gm$  stripcharts there is the solid mass of the  $gm$  values which fill the area under the similarly looking curves, descending from the highest values of  $gm$  to zero. The curve for Lena ends just before the end of the third cluster, the curve for the human brain ends just before the end of the second cluster. The set of pixels with  $gm$  under these curves was checked to consist of all pixels with the record of  $(n_1, \dots, n_7)$  equal to  $(0, 0, 0, 0, 0, 0, 0)$ . This means that the  $gm$  distances for all neighboring pixels exceed  $t_7$  for  $b = 0.2$ , i.e. for the image at Figure 2 they exceed 0.23.

The  $gm$  values of the pixels with records of  $n_1, \dots, n_7$  other than  $(0, 0, 0, 0, 0, 0, 0)$  form quills, i.e. they lie along separate disjoint descending curves, some of which are longer and correspond to larger groups of pixels. The first most left quill, which runs through clusters 1-3, corresponds to the subset consisting of 1774 pixels with  $(n_1, \dots, n_7) = (0, 0, 0, 0, 0, 0, 1)$ . Each pixel in this first quill has exactly *seven* neighbors with  $gm$  differing from the pixel's  $gm$  more than  $t_7(0.2) = 0.23$  and exactly *one* neighbor with this difference between  $t_6(0.2) = 0.19$  and  $t_7(0.2) = 0.23$ . The same concerns the first (most left) quill in each of the three  $gm$  stripcharts in Figure 1.

For the image in Figure 2 there is 256 different records of  $(n_1, \dots, n_7)$  which correspond to at least 1 pixel, but only 13 of them correspond to subsets





**Fig. 3.** Stripcharts of gradient module  $gm$  and gray level  $gl$  for GCA ordered pixels of a fragment of a human brain image at Figure 2

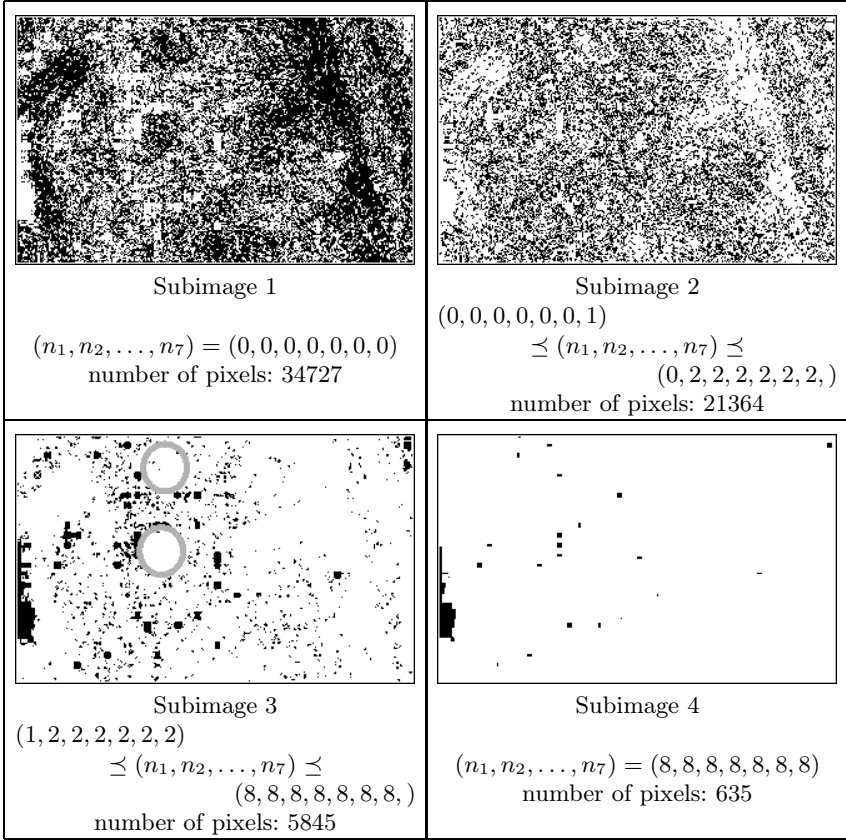
containing more than 500 pixels. The records of all these sets are indicated in the upper stripchart in Figure 3.

Each pair of quills in the  $gm$  stripchart which has ordered records of  $(n_1, \dots, n_7)$  (e.g. a pair corresponding to record  $(0, 0, 0, 0, 0, 1, 1)$  and record  $(0, 0, 0, 0, 0, 0, 2)$ ) has the sets of  $gm$  values *stochastically ordered* (i.e. for any value  $a$  of  $gm$  the fraction of  $gm$ 's not exceeding  $a$  is in the first quill not smaller than that in the second quill). This holds also when any quill is compared with the subset of pixels such that their  $(n_1, \dots, n_7)$  record is equal to  $(0, 0, 0, 0, 0, 0, 0)$ . The stochastic order of quills is visible in the upper part of Figure 3. We say that one quill is "left" to another quill if the  $(n_1, \dots, n_7)$  record for the former quill precedes the record for the latter. An analogous although less regular and clear ordering exists also for the sets of the  $gl$  values (cf. the lower stripchart in Figure 3) but this is more complicated and so will not be considered in this paper.

The subsets of pixels reordered according to their  $(n_1, \dots, n_7)$  records have *gradually less differentiated  $gm$  neighborhoods*. As shown in Section 4, it is possible to cluster some subsets of pixels with adjacent  $(n_1, \dots, n_7)$  records and then restore their spatial structure. So the image will be decomposed into subimages with gradually less differentiated neighborhoods, precisely described in relation to  $gm_{max}$ . These subimages reveal areas with homogeneous neighborhoods.

### 4 Image Decomposition into Subimages of Pixels with Adjacent Values of Neighborhood Variables

The clustering structure for the sequence of subsets of pixels with adjacent values of  $(n_1, \dots, n_7)$  can be arbitrarily chosen. For the image in Figure 2 only three



**Fig. 4.** Image from Figure 2 partitioned into subimages 1-3 containing pixels with  $n_1, \dots, n_7$  records belonging to the respective three clusters. Subimage 3 contains subimage 4. The almost empty gray circles in subimage 3 indicate the homogeneous compact areas of pixels with similar highly differentiated  $gm$ 's inside their neighborhoods.

clusters were formed: the first for pixels with  $(n_1, \dots, n_7) = (0, 0, 0, 0, 0, 0, 0)$ , the second with  $(n_1, \dots, n_7)$  between  $(0, 0, 0, 0, 0, 0, 1)$  and  $(0, 2, 2, 2, 2, 2, 2, )$ , the third with  $(n_1, \dots, n_7)$  between  $(1, 2, 2, 2, 2, 2, 2)$  and  $(8, 8, 8, 8, 8, 8, 8, )$ . The resulting subimages 1-3 are presented in Figure 4 which, additionally, contains subimage 4 related to record  $(8, 8, 8, 8, 8, 8, 8, )$ .

It is left to the reader to find out the difference between subimages 3 and 4. Clearly, there has to exist a contour line at the edge of the black area placed at the left side of subimage 3, formed by some thinner contour lines, each referring to a record placed between  $(1, 2, 2, 2, 2, 2, 2)$  and  $(8, 8, 8, 8, 8, 8, 8, )$ .

Subimage 1 contains two areas worth special attention. They lie approximately in the circles (drawn in subimage 3) with the centers situated in points  $(126, 174)$  and  $(120, 110)$  which were mentioned in Section 3. The circles in subimage 3 are

practically empty (no more than 10 pixels in the upper circle and no more than 5 in the lower circle). Contrarily, a majority of all pixels belonging to the upper and the lower circle lies in subimage 1.

A more detailed decomposition of subimage 2 is under preparation. It would enable a more precise description of the circles which could be useful in the future medical recognition and classification.

It is interesting that the two circles are almost identical in each of the three subimages. Let us suppose that both circles become identically classified by a specialist from the medical point of view. This will suggest that the description of  $gm$  neighborhoods might help to specify that particular phenomenon.

In any case the exceptional homogeneity of the  $gm$  neighborhoods which appears in both circles needs a closer investigation.

## 5 Conclusions

The decomposition described in Section 4 can be modified according to the choice of the value of parameter  $b$ , the number of neighborhood variables, the size of pixel neighborhoods, the number and sizes of clusters aggregating pixels with adjacent neighborhood records. By a diligent preliminary search throughout that parameter space at the learning stage, followed by a sequence of well done choices, it is possible to get a deep knowledge of the  $gm$  neighborhoods in particular parts of the image.

If  $b = 0$  then all neighborhood variables are equal to zero and therefore the post GCA  $gm$  stripchart consists only of the  $gm$ 's of pixels ordered according to the quotient  $gm/gl$ . If  $b$  is non-zero but very small, the fraction of pixels with record  $(0, 0, 0, 0, 0, 0, 0)$  is relatively large, and it diminishes when  $b$  increases. Thus small values of  $b$  are particularly useful to reveal thin contour lines at the edges of areas with homogeneous  $gm$  neighborhoods.

An increase of the number of neighborhood variables could be very useful to classify and describe homogeneous  $gm$  neighborhoods. This would be better for a preliminary recognition of image neighborhoods than changing the values of parameter  $b$ . Also investigating in a preliminary search the neighborhoods consisting of 24 pixels instead of 8 pixels seems highly advisable in a preliminary search.

However, the subimages in Figure 4 suggest that the formation of clusters consisting of pixels with adjacent neighborhood records is of special importance. A look on the post GCA  $gm$  stripchart might be a good guide to make a preliminary choice.

Computational effort accompanying the use of procedures described in the present paper is big. A thorough analysis of a gray level image by methods based on GCA needs a computational tool more developed and specialized than the GradeStat programme in its present form, but preserving the grade approach to the analysis.

**Acknowledgement.** Partly sponsored by grant No. 3 T11C 053 28 from the Ministry of Scientific Research and Information Technology, concerning theory, software and applications of grade multivariate analysis.

## References

1. Kowalczyk T, Pleszczyńska E, Ruland F (eds)(2004) *Grade Models and Methods for Data Analysis, With Applications for the Analysis of Data Populations*, Series: *Studies in Fuzziness and Soft Computing*, vol. 151, 477 p., Springer Verlag, Berlin Heidelberg New York
2. <http://gradestat.ipipan.waw.pl/>
3. Książczyk J, Matyja O, Pleszczyńska E, Wiech M. (eds)(2005) *Analysis of medical and demographical data aided by programme GradeStat*. Institute of Computer Science PAS, the Children's Memorial Health Institute (in Polish)
4. Pleszczyńska E (2007) *Application of grade methods to medical data: new examples*. *Biocybernetics and Biomedical Engineering*, No. 3
5. Grzegorek M (2005) *Image Decomposition by Grade Analysis - an Illustration*. In: Kurzyński M, Puchała E, Woźniak M, Żołnierek A (eds) *Computer Recognition Systems. Proceedings of the 4th International Conference on Computer Recognition Systems CORES'05*, May 22-25, Rydzyna Castle. Series: *Advances in Soft Computing*. Springer Verlag: Berlin Heidelberg New York, pp 387-394
6. Grzegorek M (2005) *Decomposition of the fragment of brain image performed using grade data methods with the aid of GradeStat program*. In: Rutkowski L (ed.), *Proc. of XIVth Scientific Conference "Biocybernetics and Biomedical Engineering"* Poland, Częstochowa, 21-23 Sept, pp 260-265. (in Polish)
7. Grzegorek M (submitted) *Decomposition of a medical image by a new tool based on grade multivariate methodology*
8. Gonzalez RC, Woods RE (2002) *Digital Image Processing. Second Edition*. Upper Saddle River, Prentice Hall

---

# Comparison of Hierarchical Cluster Analysis Methods Applied to Image Segmentation by Watershed Merging

Jakub Smolka and Maria Skublewska-Paszowska

Institute of Computer Science, Lublin University of Technology  
jakub.smolka@pollub.pl, mariap@pluton.pol.lublin.pl

**Summary.** Watershed transformation is an image segmentation method. Its main advantage is that it extracts almost all edges present in the image. The edges are always continuous, which is the method's main advantage. Unfortunately it detects all edges - even if they are very weak. This causes more classes being segmented than there are objects present in the image. This phenomenon is called over-segmentation.

This paper presents a solution to the over-segmentation problem. The solution uses hierarchical cluster analysis for joining similar classes of the over-segmented image into a given number of clusters. It treats watersheds as objects characterized by a set of attributes. Four different clustering methods, seven attributes combined in different sets, seven similarity/dissimilarity measures and four standardization methods were compared using quality assessment functions. Tests were performed on CT and MRI scans. Combinations that achieved the highest number of good segmentations are described. Selected segmentations are presented.

## 1 Introduction

Watershed transformation was introduced by Beucher and Lantuejoul in 1979 [1] and is still a popular subject of research [8, 9]. It is a segmentation method that mimics pouring water onto a landscape created on the basis of a digital image. Water raining onto a landscape flows with gravity and collects in catchment basins (also called watersheds) [2, 3]. To prevent water from spilling from one basin into another, the segmentation algorithm builds "dams" between catchment basins. This causes all the boundaries to be continuous. The algorithm assumes that higher pixel values indicate the presence of boundaries in the image  $I(x)$ . The output of the gradient operator satisfies this requirement and, for this reason, is often used for obtaining the height function  $I(x) = |\nabla f(x)|$  from the original image  $f(x)$ . The gradient operator is very sensitive to noise [7], and since the transformation produces a region for each of the image's local minima, the resulting number of regions (watersheds) is significantly larger than number of objects depicted in the image: hence the image is over-segmented. The over-segmentation is illustrated in figures 2b and 3b.

## 2 Watershed Merging by Means of the Hierarchical Cluster Analysis Methods

In the described method, watersheds are treated as objects with a set of attributes assigned. Their similarity is measured with a similarity or dissimilarity coefficient. There are a number of coefficients that can be used to perform clustering. Four clustering methods were included in this comparison: *single linkage* (SLINK), *complete linkage* (CLINK), *unweighted pair-group method using arithmetic averages* (UPGMA) and *Ward's minimum variance method*. The first three are very similar [4].

Before the clustering method can be executed, the clustering parameters need to be specified. These parameters include: attribute set, standardization method and similarity or dissimilarity coefficient. The watershed clustering process with SLINK, CLINK and UPGMA methods is straightforward: (1) attribute values are calculated for each watershed, (2) optionally the data is standardized, (3) the similarity (or dissimilarity) matrix is constructed using one of similarity/dissimilarity coefficients described below, (4) the two most similar clusters are found, (5) the clusters are merged (similarity tree which represents the clustering hierarchy is updated), (6) the similarity (or dissimilarity) matrix is updated, (7) if there are two or more clusters left, the algorithm goes back to step 4. The SLINK, CLINK and UPGMA methods differ only in the way they update the similarity (or dissimilarity) matrix. SLINK considers two clusters as similar as their two most similar components (i.e. watersheds). Conversely, CLINK considers two clusters as similar as their two most dissimilar components. UPGMA can be classified in between SLINK and CLINK because it averages the similarity measures of all possible pairs of components in the two clusters.

Ward's minimum variance method doesn't use similarity/dissimilarity coefficients. Instead, the algorithm attempts to find a merger that will cause a minimal increase in the total within-cluster error sum of squares [5]. Consequently in each iteration, the algorithm has to check all possible mergers.

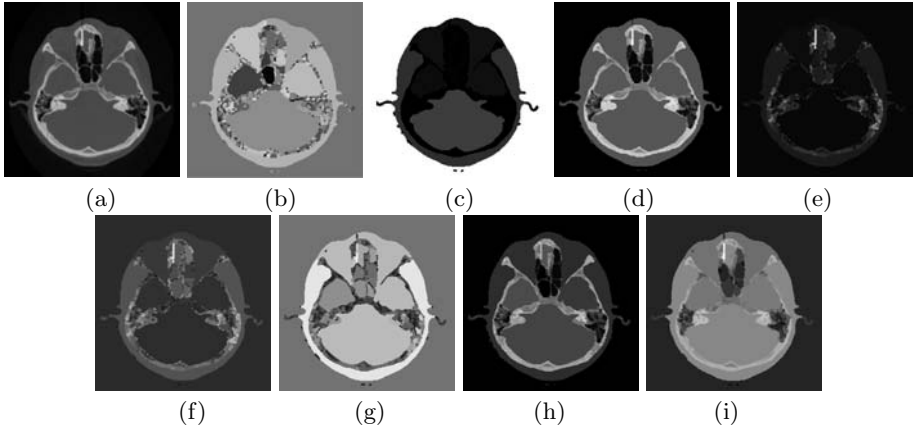
Regardless of the clustering method used, the result is a similarity hierarchy which can be represented as a tree. Each node in the tree represents a merger of two clusters and has a value (usually a measure of their similarity) associated with it. With the similarity tree available, obtaining the required number of classes comes down to picking the appropriate value of the tree node (similarity measure). This can be thought of as cutting the tree into two parts: top and bottom. The top part (containing the final clustering steps) is left out, and the bottom determines how the watersheds are merged.

### 2.1 Hierarchical Cluster Analysis Parameters

Hierarchical cluster analysis methods depend upon three types of parameters that can easily be exchanged. These parameters are: watershed attributes, standardization method and similarity/dissimilarity coefficient. In the equations below the following symbols are used:  $j, k$  - object's numbers,  $t$  - number of objects,  $n$  - number of attributes,  $X_{ij}$  - value of  $i$ -th attribute of  $j$ -th object,  $X_{ik}$  - value

of  $i$ -th attribute of  $k$ -th object,  $X_{\min i}, X_{\max i}$  - minimum and maximum values of  $i$ -th attribute,  $\bar{X}_i$  - mean value of  $i$ -th attribute,  $\sigma_i$  - standard deviation of  $i$ -th attribute.

In this comparison of clustering methods, seven different attributes were used. The first included attribute is *watershed size* (figure 1c). It is expressed as the number of pixels the watershed consists of. It was included because sometimes only selected classes are over-segmented like bone tissue in figure 2b. The second kind of attribute (figure 1d) used here is *watershed mean value* (denoted avg. in table 1). The third and the fourth attributes are *variance* (denoted var.) and *standard deviation*  $\sigma_i$  (denoted s.d.). These two attributes are similar. They are, to some extent, sensitive to image texture. Regions that are almost identical with respect to their mean value may be considered different when their variance or standard deviation is taken into account (figures: 1e and 1f). The fifth attribute is the *value spread* (figure 1g). It is also partially dependent on texture. The sixth and seventh are *minimal value* (denoted min.) and *maximal value* (denoted max.) correspondingly (figures: 1h and 1i).



**Fig. 1.** Watershed attributes: (a) original image, (b) watersheds, (c) size, (d) average, (e) variance, (f) standard deviation, (g) value spread, (h) minimal value, (i) maximal values

Attributes usually take values from different ranges. Standardization may be needed to prevent one attribute from dominating others. Four different standardization equations were used in the comparison. The first equation

$$Z_{ij} = (X_{ij} - \bar{X}_i) / \sigma_i \quad (1)$$

(denoted S in table 1) makes the standardized attribute have a mean value of  $\bar{Z}_i = 0$  and a standard deviation of  $\sigma_i = 1$  [4]. The next two equations

$$Z_{ij} = X_{ij} / X_{\max i} \quad (2)$$

(denoted L1) and

$$Z_{ij} = (X_{ij} - X_{\min i}) / (X_{\max i} - X_{\min i}) \tag{3}$$

(denoted L01) scale the data [4] to the range of [0,1]. But only in the case of the second equation, is the lowest value guaranteed to be equal to 0. The last equation

$$Z_{ij} = X_{ij} / \sum_{i=1}^t X_{ij} \tag{4}$$

(denoted LS1) causes the normalized data to sum up to 1 [4].

There are many different similarity and dissimilarity measures that can be used in combination with clustering methods. Seven such measures have been used for grouping watersheds. Only the measures that were found useful are described in more detail. The first dissimilarity measure is the *Euclidean distance coefficient* [4] given by

$$e_{jk} = \sqrt{\sum_{i=1}^n (X_{ij} - X_{ik})^2} \tag{5}$$

It represents the distance between two points in n-dimensional space. The *average Euclidean distance coefficient* [4]

$$d_{jk} = \sqrt{\sum_{i=1}^n (X_{ij} - X_{ik})^2 / n} \tag{6}$$

differs from  $e_{jk}$  in that it is able to compensate for missing values by using the scaling factor  $1/n$ . The *Canberra metric coefficient* [4]

$$a_{jk} = \frac{1}{n} \sum_{i=1}^n |X_{ij} - X_{ik}| / (X_{ij} + X_{ik}) \tag{7}$$

equalizes the contribution of each attribute to overall similarity. The last dissimilarity measure presented here is the *Bray-Curtis coefficient* [4]

$$b_{ij} = \left( \sum_{i=1}^n |X_{ij} - X_{ik}| \right) / \left( \sum_{i=1}^n (X_{ij} + X_{ik}) \right) \tag{8}$$

Unlike the Canberra metric, it allows one attribute to be dominant. Other similarity/dissimilarity measures included in the comparison are: *the coefficient of shape difference*, *the cosine coefficient* and *the correlation coefficient* (also known as the Pearson product-moment correlation coefficient) [4]. These coefficients can be used for obtaining good segmentation but usually didn't give good results. In some cases it was impossible to calculate them for parts of the test images.



### 3 Clustering Methods Comparison

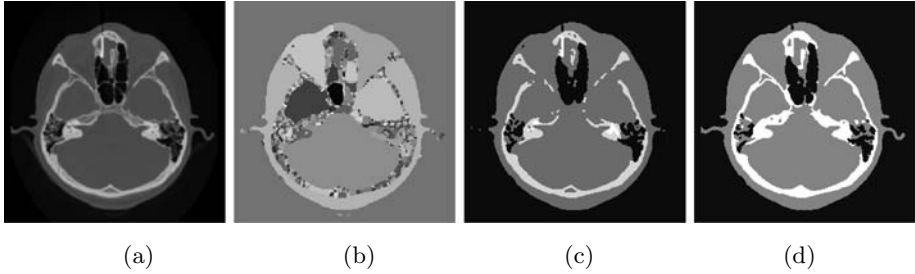
In the comparison of hierarchical cluster analysis methods four different clustering algorithms were included: SLINK, CLINK, UPGMA and Ward's minimum variance method. Each method was tested with four standardization equations plus one test with no standardization. Seven similarity/dissimilarity coefficients were used with SLINK, CLINK and UPGMA methods. Ward's method has a built-in similarity measure; hence, it couldn't be used with the coefficients described above. Twelve attribute sets were tested: (1) watershed's average, (2) watershed's average and standard deviation, (3) watershed's average, standard deviation and size, (4) watershed's average and variance, (5) watershed's average, variance and size, (6) watershed's minimal value, (7) watershed's maximal value, (8) watershed's average and value spread, (9) watershed's average and minimal value, (10) watershed's average and maximal value, (11) watershed's average and both minimal and maximal value, (12) watershed's average, minimal value, maximal value and the value spread. All possible combinations of method/standardization equation/attribute set were tested on four test images: CT, MRI PD, MRI T1 and MRI T2 scans. All test images were preprocessed with a gradient diffusion filter [6]. The gradient filter was applied to extract edges. The resulting image was in turn thresholded with a threshold value equal to 5% (for CT) or 2% (for MRI) of maximal brightness in the gradient image. The required number of classes was between 2 and 12 classes resulting in eleven segmentations being created in each case. This gave a total of 46,464 segmentations created. The quality of each segmentation was assessed using five quality measures:

1. mean value of the difference image:  $a_d = \sum_{x \in I} |I_d(x)| / n$ ,
2. mean square error:  $MSE = \sum_{x \in I} (I(x) - I_a(x))^2 / n$ ,
3. mutual information:  $M_{I,I_a} = \sum_{i=0}^k \sum_{j=0}^m p(i, j) \cdot \log p(i, j) / (p_I(i) \cdot p_{I_a}(j))$ ,
4. standard deviation of the difference image:  $\sigma_d = \sqrt{\sum_{x \in I} \frac{(I_d(x) - \bar{I}_d)^2}{(n-1)}}$ ,
5. redundancy:  $R_{I,I_a} = M_{I,I_a} / (H_I + H_{I_a})$ ,

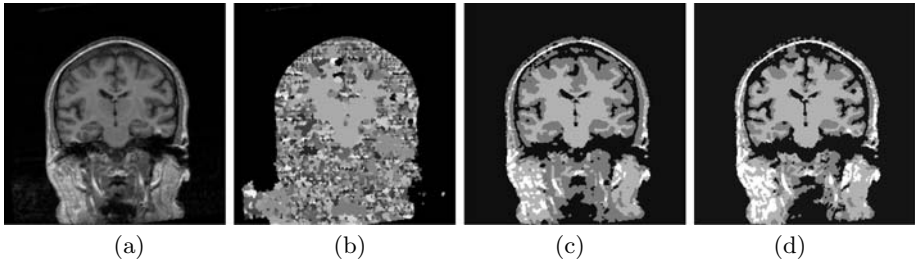
where:  $I$  - original image,  $I_a$  - image with classes' average values,  $I_d$  - difference image,  $\bar{I}_d$  - mean value of difference image,  $x$  - a pixel,  $n$  - pixel count,  $k$  - number of different gray levels in the original image,  $m$  - number of classes in the final segmentation,  $p_I, p_{I_a}$  - pixel value probability distributions in image  $I$  and  $I_a$ ,  $p$  - joint probability distributions of gray levels in images  $I$  and  $I_a$ ,  $H_I, H_{I_a}$  - entropies of  $I$  and  $I_a$ .

Once all the segmentations had been performed, their quality was assessed using the above-mentioned quality assessment functions. Then, for a particular number of classes (ranging from 2 to 12), the highest values of all assessment functions were found. A particular segmentation was considered good when at least one of the quality measures had a value differing by no more than 10%

from the best segmentation. Segmentation parameters (method, standardization equation and attribute set) were ordered by the number of cases in which they produced a good segmentation (regardless of the number of classes). The best segmentation parameters are shown in Table 1. Two example segmentations are depicted in figure 2.



**Fig. 2.** CT scan segmentation (the scan comes from The Visualization Toolkit's example data package) (a) original image, (b) over-segmented image (885 watersheds), (c) UPGMA with Euclidean distance coefficient, no standardization, 4 classes, attributes: average, standard deviation, (d) CLINK with Bray-Curtis coefficient, no standardization, 3 classes, attributes: average, minimal and maximal values



**Fig. 3.** MRI scan segmentation (MR brain data set 788\_6\_m was provided by the Center for Morphometric Analysis at Massachusetts General Hospital and is available at <http://www.cma.mgh.harvard.edu/ibsr/>) (a) original image, (b) over-segmented image (2907 watersheds), (c) Ward's method, no standardization, 4 classes, attributes: average, standard deviation, (d) Ward's method, no standardization, 4 classes, attributes: average, maximal value

## 4 Conclusions

The presented results show that these hierarchical cluster analysis methods can be used for watershed merging. The results presented in table 1 show that optimal segmentation parameters depend upon the type of the image. However, some general conclusions may be drawn. UPGMA gives the best results for CT images. It also generates good segmentations in case of the MRI scans but not as

**Table 1.** Segmentation parameters that produced highest number of good segmentations

		CT			MRI PD			
Pos.	Method	Coefficient	Std.	Attr. set	Method	Coefficient	Std.	Attr. set
1	UPGMA	Euclidean	none	avg. s. d.	WARD	none	none	avg. min.
2	UPGMA	avg. Eucl.	none	avg. s. d.	WARD	none	none	avg. s. d.
3	CLINK	Bray-Curtis	none	avg. min.	WARD	none	L1	avg. var.
4	CLINK	Bray-Curtis	L1	avg. var.	WARD	none	L01	avg. var.
5	CLINK	Euclidean	none	avg.	WARD	none	LS1	avg.
6	CLINK	Euclidean	L1	avg.	WARD	none	S	avg.
7	CLINK	Euclidean	L01	avg.	CLINK	Euclidean	none	avg.
8	CLINK	Euclidean	LS1	avg.	CLINK	Euclidean	L1	avg.
9	CLINK	Euclidean	LS1	avg. min.	CLINK	Euclidean	L01	avg.
10	CLINK	Euclidean	S	avg.	CLINK	Euclidean	S	avg.

		MRI T1			MRI T2			
Pos.	Method	Coefficient	Std.	Attr. set	Method	Coefficient	Std.	Attr. set
1	CLINK	Euclidean	none	avg. max.	WARD	none	none	avg.
2	CLINK	Euclidean	LS1	avg.	WARD	none	none	avg. s. d.
3	CLINK	avg. Eucl.	none	avg. max.	WARD	none	L1	avg.
4	CLINK	avg. Eucl.	LS1	avg.	WARD	none	L01	avg.
5	Ward's	none	none	avg. min.	WARD	none	S	avg.
6	WARD	none	none	avg.	CLINK	Bray-Curtis	none	avg.
7	WARD	none	none	avg. s. d.	CLINK	Bray-Curtis	L1	avg.
8	WARD	none	L1	avg.	CLINK	Bray-Curtis	LS1	avg.
9	WARD	none	L01	avg.	CLINK	Canberra	none	avg.
10	CLINK	Euclidean	none	avg.	CLINK	Canberra	L1	avg.

frequently as CLINK and Ward’s methods. As a consequence, UPGMA was not among the best parameter sets for MRI scans. Ward’s method gave best results in the case of MRI PD and MRI T2 scans, whereas CLINK was most successful when used for merging watersheds in the over-segmented MRI T1 scan. SLINK failed in almost all cases.

In most successful segmentations, methods that need a similarity measure (that is CLINK or UPGMA) were used in combination with the Euclidean distance, average Euclidean distance or Bray-Curtis coefficients. The Canberra metric produced it’s share of good results. Other coefficients failed frequently. Ward’s method does not use an “external” similarity measure.

Attribute sets used in most successful segmentations differ from each other. However several things should be noted: (1) in the case of the images other than MRI T2, attribute sets usually contain more than one attribute. This means that adding an extra attribute can improve segmentation quality, (2) watershed size and value spread do not appear to be relevant when merging watersheds in CT or MRI scans, (3) standardization is not always necessary however it improves the segmentation quality when attributes that differ in the order of magnitude are used (i.e. average and variance). The following attribute sets allowed for obtaining good segmentations: (1) average and standard deviation, (2) average

and minimal value, (3) average and maximal value, (4) average, (5) average and variance and finally (6) average, the minimal and maximal values.

## References

1. Beucher S, Lantuejoul C (1979) Use of watersheds in contour detection. International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation
2. Beucher S (1992) Scanning Microscopy International 6:299–314
3. Ibanez L, Schroeder W, Ng L, Cates J (2003) The ITK Software Guide. Kitware Inc
4. Romesburg Ch (2004) Cluster Analysis for Researchers. Lulu Press, North Carolina
5. Everitt B S, Landau S, Leese M (2001) Cluster Analysis. Arnold a member of the Hodder Headline Group, London
6. Perona P, Malik J (1990) IEEE Transactions on Pattern Analysis Machine Intelligence 12:629–639
7. Russ J C (2002) The Image Processing Handbook. CRC Press, Boca Raton London New Your Washington D.C.
8. Malpica N, Ortuño J E, Santos A (2003) Pattern Recognition Letters 24:1545-1554
9. Grau V, Raya M A, Monserrat C, Juana M C, Martí-Bonmatí L (2004) Pattern Recognition 37:47-59

---

# Modified Diffusion to Image Denoising

Dariusz Borkowski

Faculty of Mathematics and Computer Science, Nicolaus Copernicus University,  
ul. Chopina 12/18, 87-100 Toruń, Poland  
dbor@mat.uni.torun.pl

**Summary.** In this paper a novel approach for image denoising using stochastic differential equations (SDEs) is presented. In proposed method a controlled parameter to Euler's approximations of solutions to SDEs with reflecting boundary is added. It is shown that modified diffusion gives very good results for Gaussian noise source models and compares favourably with other image denoising filters.

## 1 Introduction

Let  $D$  be a bounded, convex domain in  $\mathbb{R}^2$ ,  $u : \overline{D} \rightarrow \mathbb{R}$  be an original image and  $u_0 : \overline{D} \rightarrow \mathbb{R}$  be the observed image of the form

$$u_0 = u + \eta,$$

where  $\eta$  stands for a white Gaussian noise. We assume that  $u$  and  $u_0$  are appropriately regular. We are given  $u_0$ , the problem is to reconstruct  $u$ .

The inverse problem of restoration of noisy image by automatic and reliable methods belongs to the most intensively studied topics of image processing. Various techniques of noise removal were proposed to tackle this problem. One may quote the linear filtering, variational/PDE-based approaches [1, 2, 4, 5], wavelets theory and stochastic modelling (generally based on the Markov field theory). In this paper we study the restoration based on reflected stochastic differential equations. We will present a new stochastic method: modified diffusion, which involves advanced tools of stochastic analysis.

## 2 Mathematical Preliminaries

Let  $D \subset \mathbb{R}^n$  be a domain with closure  $\overline{D}$  and boundary  $\partial D$ . Let  $T > 0$  and by  $\mathbb{C}([0, T]; \mathbb{R}^n)$  denote the set of  $\mathbb{R}^n$  valued continuous functions  $f : [0, T] \rightarrow \mathbb{R}^n$ .

**Definition 1.** Let  $y$  be an element of the set  $\mathbb{C}([0, T]; \mathbb{R}^n)$  such that  $y_0 \in \overline{D}$ . A pair  $(x, k) \in \mathbb{C}([0, T]; \mathbb{R}^{2n})$  is said to be a solution to the Skorokhod problem associated with  $y$  and  $D$  if

- (i)  $x_t = y_t + k_t, \quad t \in [0, T],$
- (ii)  $x_t \in \overline{D}, \quad t \in [0, T],$

(iii)  $k$  is a function with bounded variation  $|k|$  on  $[0, T]$ ,  $k_0 = 0$  and

$$k_t = \int_0^t n_s d|k|_s, \quad |k|_t = \int_0^t 1_{\{x_s \in \partial D\}} d|k|_s, \quad t \in [0, T],$$

where  $n_s = n(x_s)$  is a normal unit vector at  $x_s \in \partial D$ .

It is known that if set  $D$  is convex, there exists a unique solution to the Skorokhod problem (see [7]).

**Definition 2.** Let  $(\Omega, \mathcal{F}, \mathcal{P})$  be a probability space.

(i) A stochastic process is a parametrised collection of random variables  $X = \{X_t; t \in [0, T]\}$  defined on a probability space  $(\Omega, \mathcal{F}, \mathcal{P})$  with values in  $\mathbb{R}^n$ .

For each fixed  $\omega \in \Omega$  the function  $t \rightarrow X_t(\omega)$  is called a trajectory of  $X$  and is denoted by  $X(\omega)$ .

(ii) A filtration  $(\mathcal{F}_t) = \{\mathcal{F}_t; t \in [0, T]\}$  is a nondecreasing family of sub- $\sigma$ -fields of  $\mathcal{F}$  i.e.  $\mathcal{F}_s \subseteq \mathcal{F}_t \subseteq \mathcal{F}$  for  $0 \leq s < t \leq T$ .

$(\mathcal{F}_t^X)$  denotes a filtration generated by process  $X$ , i.e.  $\mathcal{F}_t^X = \sigma(X_s; 0 \leq s \leq t)$ .

(iii) A stochastic process  $X$  is adapted to the filtration  $(\mathcal{F}_t)$  ( $X$  is  $(\mathcal{F}_t)$  adapted) if for each  $t \in [0, T]$ ,  $X_t$  is  $\mathcal{F}_t$  - measurable random variable.

**Definition 3.** Let  $Y$  be an  $(\mathcal{F}_t)$  adapted process with continuous trajectories,  $Y_0 \in \overline{D}$ . We say that a pair  $(X, K)$  of  $(\mathcal{F}_t)$  adapted processes is a solution to the Skorokhod problem associated with  $Y$  and  $D$ , if for almost every  $\omega \in \Omega$ ,  $(X(\omega), K(\omega))$  is a solution to the Skorokhod problem associated with  $Y(\omega)$  and  $D$ .

In what follows, by  $W$  denote a Wiener process,  $x_0 \in \overline{D}$  and  $\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^m$ .

**Definition 4.** Let  $Y$  be an  $(\mathcal{F}_t)$  adapted process. A pair  $(X, K^{\overline{D}})$  of  $(\mathcal{F}_t)$  adapted processes is called a solution to reflected SDE

$$X_t = x_0 + \int_0^t \sigma(X_s) dW_s + K_t^{\overline{D}}, \quad t \in [0, T], \quad (1)$$

if  $(X, K^{\overline{D}})$  is a solution to the Skorokhod problem associated with

$$Y_t = x_0 + \int_0^t \sigma(X_s) dW_s, \quad t \in [0, T]$$

and  $D$ .

The proof of existence and uniqueness of the solution to SDEs with reflection can be found in [7].

### 3 Stochastic Representation of Solution to the Heat Equation

Before presenting a general method, we will illustrate our ideas by constructing a model which is equivalent to a commonly used filter, namely, the convolution of the noise image with two-dimensional Gaussian mask. The construction of our model is based on construction of the process  $X$ . We suppose for a while that the image is given by a function defined on the whole plane and put  $X_t = W_t^x$ ,  $t \in [0, T]$ , where  $W^x$  is a Wiener process starting from  $x \in \overline{D}$ .

Then

$$\mathbf{E}[u_0(X_T)] = \int_{\mathbb{R}^2} G_{\sqrt{T}}(x - y)u_0(y) dy, \tag{2}$$

where  $G_\gamma(x) = \frac{1}{2\pi\gamma^2}e^{-\frac{|x|^2}{2\gamma^2}}$  is two-dimensional Gaussian mask.

The mean value defined above is the reconstructed pixel  $u(x)$ . Therefore, the image is the convolution of the noise image with two-dimensional Gaussian mask.

While discussing the above example, we assumed that the image is the function given on the whole plane. Since we want to consider the image as a function defined on the bounded convex set, we have to introduce a new assumption on the process  $X$ . Suppose that the process  $X$  is a stochastic process with reflection with values in  $\overline{D}$ . In this case process  $X$  is given by a Wiener process with reflection, i.e. it can be written as  $X_t = W_t^x + K_t^{\overline{D}}$ .

### 4 Image Denoising

The model constructed in the previous section is equivalent to the convolution of the noise image with two-dimensional Gaussian mask. This filter removes noise and blurs edges. In this section we provide a construction [3] with the following properties: noise is removed and image has sharp edges.

In a neighbourhood of an edge, the image exhibits a strong gradient. In order to preserve this edge, we should diffuse along it [5]. Assume that the process  $X$  has the form

$$X_t = x + \int_0^t \sigma_1(X_s) dW_s + K_t^{\overline{D}}, \tag{3}$$

where

$$\sigma_1(X_t) = \lambda_1(|\nabla u_0(X_t)|) \begin{bmatrix} -\frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|}, 0 \\ \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|}, 0 \end{bmatrix}, \lambda_1(s) > 0.$$

At locations where the variations of the brightness are weak (low gradient), we would like to encourage smoothing, the same in all direction. We expect that process  $X$  will have the property of the Wiener process. This condition may be achieved by imposing

$$X_t = x + \int_0^t \sigma_2(X_s) dW_s + K_t^{\overline{D}}, \tag{4}$$

where the diffusion coefficient has the form

$$\sigma_2(X_t) = \lambda_2(|\nabla u_0(X_t)|) \begin{bmatrix} -\frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|}, \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|} \\ \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|}, \frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|} \end{bmatrix}, \quad \lambda_2(s) > 0.$$

Combining the assumptions (3) and (4) we can write  $X$  as

$$X_t = x + \int_0^t \sigma(X_s) dW_s + K_t^{\overline{D}}, \quad (5)$$

where

$$\sigma(X_t) = \begin{bmatrix} -\lambda_1(|\nabla u_0(X_t)|) \frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|}, \lambda_2(|\nabla u_0(X_t)|) \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|} \\ \lambda_1(|\nabla u_0(X_t)|) \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|}, \lambda_2(|\nabla u_0(X_t)|) \frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|} \end{bmatrix},$$

$$\lim_{s \rightarrow 0} \lambda_1(s) > 0, \lim_{s \rightarrow 0} \lambda_2(s) > 0, \lim_{s \rightarrow 0} \frac{\lambda_1(s)}{\lambda_2(s)} = 1, \lim_{s \rightarrow \infty} \lambda_1(s) > 0, \lim_{s \rightarrow \infty} \lambda_2(s) = 0.$$

## 5 A Numerical Scheme

Consider the following numerical scheme

$$\begin{aligned} X_0^m &= X_0, \\ X_{t_k}^m &= \Pi_{\overline{D}}[X_{t_{k-1}}^m + \sigma(X_{t_{k-1}}^m)(W_{t_k} - W_{t_{k-1}})], \quad k = 1, 2, \dots, m, \end{aligned} \quad (6)$$

where  $t_k = \frac{kT}{m} = kh$ ,  $k = 0, 1, \dots, m$  and  $\Pi_{\overline{D}}(x)$  denotes a projection of  $x$  on the set  $\overline{D}$ .

**Theorem 1.** *Let  $(X, K^{\overline{D}})$  be the solution to the reflected SDE (1). If there exists  $C > 0$  such that*

$$\|\sigma(x) - \sigma(y)\|^2 \leq C|x - y|^2,$$

then

$$\lim_{m \rightarrow +\infty} |X_T^m - X_T| = 0 \quad \mathcal{P} - a.s.$$

The proof of the above theorem can be found in [6].

## 6 Modified Diffusion

Recall that the mean value  $\mathbf{E}[u_0(X_T^m)]$  is the reconstructed pixel  $u(x)$ . The numerical scheme (6) gives good results, but only with small value of the parameter  $h = \frac{T}{m}$  (for example  $h = 0.01$ ). Calculating the mean value using Monte Carlo method for small  $h$  is not effective and takes a long time. To solve this problem, we improve the scheme (6).



$$\begin{aligned}
X_0^m &= X_0, & H_{t_k}^m &= \Pi_{\overline{D}}[X_{t_{k-1}}^m + \sigma(X_{t_{k-1}}^m)(W_{t_k} - W_{t_{k-1}})], \\
X_{t_k}^m &= \begin{cases} H_{t_k}^m, & \text{if } |u_0(H_{t_k}^m) - u_0(X_{t_{k-1}}^m)| \leq p, \\ X_{t_{k-1}}^m, & \text{if } |u_0(H_{t_k}^m) - u_0(X_{t_{k-1}}^m)| > p, \end{cases} & k &= 1, 2, \dots, m. \end{aligned} \quad (7)$$

Parameter  $p$  guarantees that, if the image exhibits a strong gradient then the process  $X^m$  diffuses as a process with small value of the parameter  $h$ ; at locations where variations of the brightness are weak, the process  $X^m$  can diffuse with large value of  $h$  (for example  $h = 2$ ). For  $p = +\infty$  (in practice  $p > 255$ ) the numerical scheme (7) is equivalent with the scheme (6).

It should be observed that the numerical scheme (7) works well only, if the model of the digital image  $u_0$  is continuous. In practice, we use linear interpolation to get value of the image  $u_0$ , for any point  $x \in \overline{D}$ .

## 7 Comparison and Experimental Results

Some results from our evaluation experiments regarding the modified diffusion method and classic PDE-based methods (minimal surfaces [2], Perona-Malik [5], total variation [4]) are presented in tabular form in Table 1. In this table the maximum values of the SNR are given. The results refer to two images, *bridge* and *boat*, corrupted with the various Gaussian noise with standard deviation  $\rho$ . In all cases the modified diffusion method performs better.

Table 1. SNR

Image	<i>bridge</i>	<i>bridge</i>	<i>bridge</i>	<i>boat</i>	<i>boat</i>	<i>boat</i>
Method \ Standard deviation	$\rho = 40$	$\rho = 60$	$\rho = 80$	$\rho = 40$	$\rho = 60$	$\rho = 80$
Minimal surfaces	11.93	10.08	8.86	12.94	11.13	9.72
Perona-Malik	12.70	11.02	9.96	13.70	11.90	10.62
Total variation	12.78	11.09	10.02	13.93	12.26	10.91
Modified diffusion	<b>12.80</b>	<b>11.19</b>	<b>10.13</b>	<b>14.02</b>	<b>12.48</b>	<b>11.06</b>

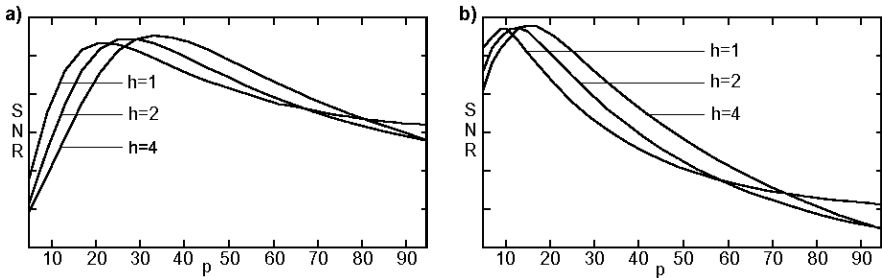
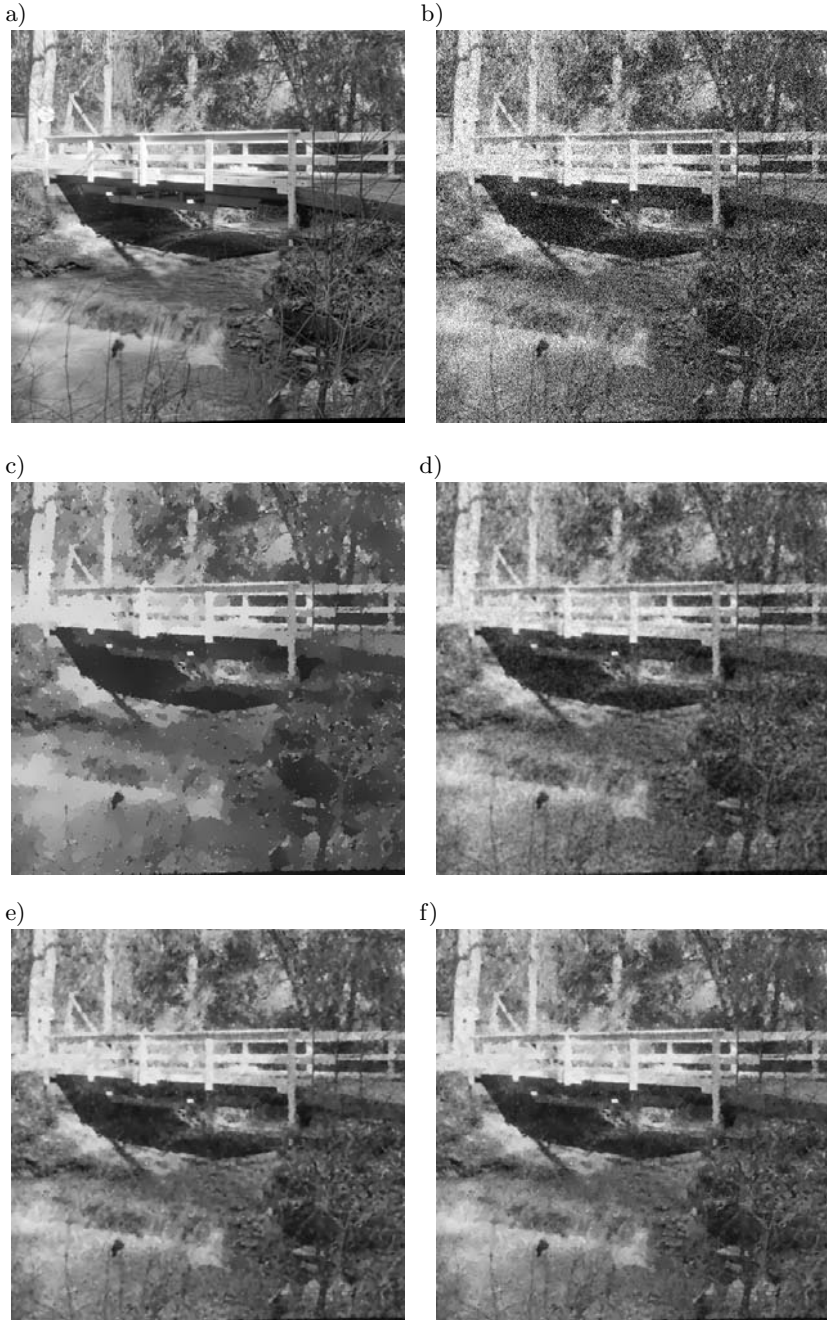
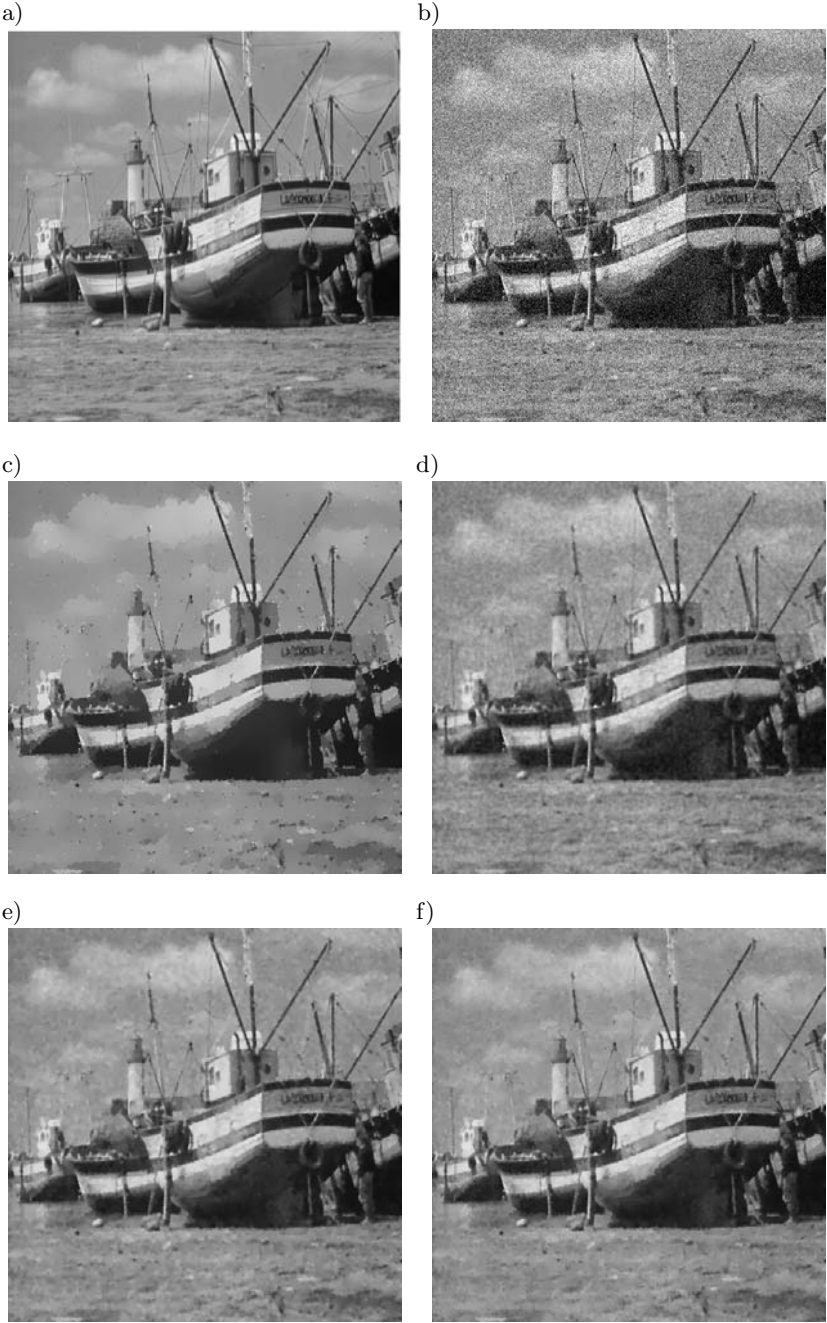


Fig. 1. SNR as a function of  $p$  for  $T = 40$  a) boat image ( $\rho = 80$ ) b) boat image ( $\rho = 40$ )



**Fig. 2.** Result of the reconstruction of the *bridge* image a) Original image b) Noisy image ( $\rho = 80$ ) c) Minimal surfaces d) Perona–Malik e) Total variation f) Modified diffusion ( $p = 15, h = 2, m = 20$ )



**Fig. 3.** Result of the reconstruction of the *boat* image a) Original image b) Noisy image ( $\rho = 60$ ) c) Minimal surfaces d) Perona–Malik e) Total variation f) Modified diffusion ( $p = 28, h = 2, m = 20$ )

In case images with strong noise ( $\rho = 80$ ) the best values of the SNR are obtained for  $p \in [20, 40]$  and  $h \in [1, 4]$ . In case when the image has weak noise ( $\rho = 40$ ) the SNR has maximum values for  $p \in [5, 20]$  and  $h \in [1, 4]$ . The values SNR as a function of parameter  $p$  are shown in Fig. 1.

In Fig. 2 and Fig. 3 we present denoising of the *bridge* and *boat* images. We add Gaussian noise to the original image with standard deviation  $\rho = 60$  for *bridge* image and  $\rho = 80$  for *boat* image. Images have been denoised by running minimal surfaces method, Perona-Malik method, total variation method and modified diffusion.

## References

1. Gilles Aubert and Pierre Kornprobst (2002). Mathematical problems in image processing. Volume 147 of Applied Mathematical Sciences, Springer-Verlag, New York.
2. Gilles Aubert and Michel Barlaud and Pierre Charbonnier and Laure Blanc-Féraud (1994). Two Deterministic Half-Quadratic Regularization Algorithms for Computed Imaging. In: Proceedings of the International Conference on Image Processing. volume II, pages 168–172.
3. Dariusz Borkowski (2007). Chromaticity Denoising using Solution to the Skorokhod Problem. In: Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems, CMA, Oslo, August 8–12, 2005. Editors: X.-C. Tai, K.-A. Lie, T.F. Chan, and S. Osher. Series: Mathematics and Visualization, Springer Verlag, p. 149–161.
4. Emad Fatemi and Leonid I. Rudin and Stanley Osher (1992). Nonlinear total variation based noise removal algorithms. In: Proceedings of the eleventh annual international conference of the Center for Nonlinear Studies on Experimental mathematics : computational issues in nonlinear science. Physica D: Nonlinear Phenomena, Volume 60, Issue 1-4, p. 259-268.
5. Jitendra Malik and Pietro Perona (1990). Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7):629–639.
6. Leszek Słomiński (2001). Euler’s approximations of solutions of SDEs with reflecting boundary. Stochastic Processes and Their Applications, 94:317-337.
7. Hiroshi Tanaka (1979). Stochastic differential equations with reflecting boundary condition in convex regions. Hiroshima Math. J., 9(1):163–177.

---

# New Results in Generation of Accurate 3D Multiview Representation of Monotonous Polyhedrons

Wojciech S. Mokrzycki<sup>1</sup> and Andrzej Salamończyk<sup>2</sup>

<sup>1</sup> Faculty of Mathematics and Informatics, University of Warmia and Mazury, Zolnierska 47, Olsztyn, Poland  
mokrzycki@matman.uwm.edu.pl

<sup>2</sup> Institute of Informatics, University of Podlasie, Sienkiewicza 51, Siedlce, Poland  
andrzej@ii.ap.siedlce.pl

**Summary.** The paper presents exact 3D multiview representation of monotonous polyhedrons (on the basis of their  $B_{rep}$ ) obtained by means of faces surrounding scanning algorithm. The algorithm uses the viewing sphere and perspective projection to describe the movement of points. (called *K-M view space model*). Visual identification is done based on these multiview models and a scene depth map. The results of tests for this algorithm for some solids are also presented.

## 1 Introduction

Visual identification is based on the multiview models described in this paper and a scene depth map. The following steps, which constitute an element of strategy are usually taken in order to recognize a given system:

1. Determination of recognizable object types.
2. Definition of identification task.
3. Generation of viewing models for each object that should be identified
4. Creation of database with all views of all models.
5. Acquisition of scene space data and visual data.
6. Isolation of scene elements and transformation of those elements to model structures stored in the database.
7. Identification of objects by comparing them with database models.

Methods of generating 3D multiview representation of polyhedron for object visual identification are described in several papers, e.g. [1] ÷ [13]. Some of them (e.g., described in [2], are based on the following idea, which can be described by a series of steps. First, central views corresponding to object features chosen for identification are generated. Next, *single-view areas* are calculated on the *view sphere*. They correspond to views generated earlier. Then, it is checked if the whole viewing sphere is covered with single-view areas. If this is completed, generation of viewing representation is finished. If not, then additional views corresponding to uncovered areas of viewing sphere are generated and it is checked

again if the process is completed. Complete viewing sphere cover with single-view areas means that generated representation is complete. Completeness of the view representation means that all different views (aspects) of the object are determined for each point on the view of the sphere and a view is defined.

Papers [5, 6, 8] present better methods. Complete representation is obtained by covering the viewing sphere precisely with single-view areas without loop, but in spiral way and controlling „edge” register (of not covered area). When the register is set to „empty” the generation of representation is completed. The representation used is complete, which results from the generation method used. However, to achieve complete representation you must calculate single-view areas on viewing sphere and carry them out in a given order. Without single-view areas it is not possible to find a complete set of views. All described methods produce convex polyhedron views only.

Another group of methods in [7, 10, 12] does not use any single-view areas and is based on other principles. It consists of the following steps. A **complementary cone** (s.c. *scanning cone*) rotates around faces normals and records every *visual event* to obtain all views. This event occurs as a result of a new normal vector entering the scanning cone or by disappearance of a vector. The outcome of such a routine is a set of vectors faces that can be seen from the view sphere. They were tested for convex polyhedrons only, [12].

In [13] we presented a method (the algorithm and its implementation) based on another idea: a **complementary cone** does not rotate around faces normals, but around faces alone, fig. 1b and records every *visual event* to obtain all views. The algorithm then generates - first of all - exact 3D views (not only for convex polyhedrons) but also for monotonous polyhedrons (a class of non-convex polyhedrons). However, this implementation does not include shadows in the views. Presented method deals with these problems.

In sec. 3 of this paper we present a method for generating a viewing representation of polyhedrons, which improves and develops our previous method for monotonous polyhedrons, fig. 1a and we also present the resulting views for some solids.

## 2 Research Assumptions

We focus on developing the method and algorithm in [13] for generation of exact 3D multiview representation for monotonous polyhedrons taking shades into consideration. We use viewing sphere with perspective projection (*K-M view model*, [2]) in order to generate the representation. The following conditions have to be met to accomplish it:

1. Models are accurate - every model is equivalent to  $B_{rep}$  model.
2. Models are viewing models - to identify a given object from any view.

We consider polyhedrons monotonous and not transparent. Use of a viewing sphere as a projection space allows simple view standardization.

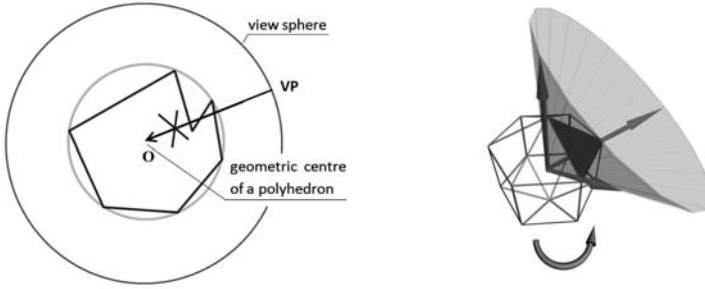


Fig. 1. Monotonous polyhedron and scanning face surrounding

### 3 Views by Scanning of Faces Surrounding

The idea of presented  $MS^{UN}$  algorithm is an extension and generalization of  $FM^{\cap V}$  algorithm from [10] generating view representation for convex polyhedrons based on  $V_{rep}$  representation. This generalization applies to scanning faces using complementary cone and tracing appearances and disappearances of faces from view.

The new scanning is performed by rotating the complementary cone around faces (not normals) and is conducted in the following way:

Let  $k_1, k_2, \dots, k_m$  be lateral edges of  $N_{rep}$  pyramid with face  $f_i$ , and let  $\gamma_1, \gamma_2, \dots, \gamma_m$  be internal angles between planes determining the edges  $k_j$  of the pyramid. We also assume that angle of flare of  $cc$  is larger than the angle of flare of every possible pyramid in  $N_{rep}$ .

Now, let us rotate  $cc$  around lateral edges of  $N_{rep}$ . It should be done in such a way that  $cc$  surface is tangent to lateral edges. We rotate the complementary cone in a range which does not allow the pyramid to which the selected edge belongs leave  $cc$ , fig. 1b. During rotation (space scanning) information is collected about polyhedron faces that enters and leaves  $cc$ , and about path of the  $cc$  axis on view sphere.

Let  $cc$  be tangent to  $k_m$  and  $k_1$  lateral edges of the first polyhedron face  $f_1$ . Let us rotate  $cc$  on  $\beta_1 = \pi - \gamma_1$  around the  $k_1$  edge - to the position of tangency with  $k_2$  (and  $k_1$ ). During rotation every visual event and  $cc$  axis path on  $VS$  should be noted.

Next,  $cc$  on  $\beta_2 = \pi - \gamma_2$  around the  $k_2$  edge is rotated - to the position of tangency with  $k_3$  and every visual event and path is noted until the rotation around  $k_m$  is completed and every visual event and path has been noted.

Then, structure of used views faces is tested and additional view in central position of scanned face should be noted if internal faces( not bordering with view contour) do not appear in the tested view.

For multiface views ( $n \geq 3$  faces) the structure of the view on the concavity seen is analyzed and the area and we find contours of concavity, and analyze overshadowing in every view.

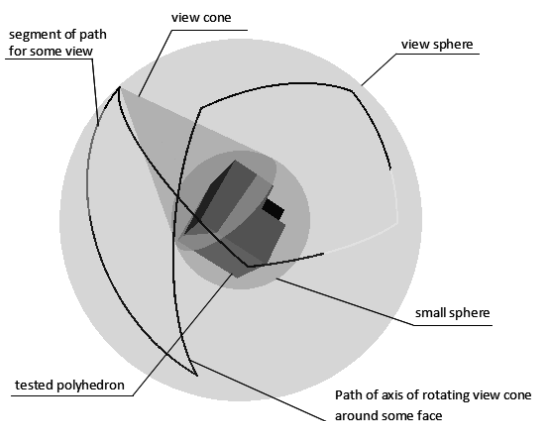
In subsequent steps, we generate views (and note paths) for the rest of polyhedron faces, test their structure and complete views and shadows.

The following events may occur:

- $cc$  includes faces together with their normals; they create a set of visible faces (at a particular view).
- $cc$  includes faces without their normals; they are invisible in the view; some of potentially visible faces can be occluded by the invisible ones.
- Faces with edges outside the cone are invisible regardless of the state of their normal vector; they belong to opposite site of polyhedron.

During scanning the edge's vectors leave end enter  $cc$ . Such events may result in new **visual events** (new view appears). Several different events may appear. Any of them may lead to a different situation.

- Covering last edge vector of some face with  $cc$  causes potential extension of view. However face could remain invisible or became invisible.
- Covering normals that are currently in  $cc$  makes these faces potentially visible, however they could still remain occluded.
- Covering faces with normals outside  $cc$  makes these faces invisible. These faces may occlude other faces that are currently immersed.
- Face may enter  $cc$  but its normal does not. It causes overshadow.
- If any edge vector of a given face leaves  $cc$ , this face leaves the actual view.



**Fig. 2.** Concept of scanning of face surrounding and path of the scanning

If an event occurs during scanning it must be carefully checked whether the actual view includes immersion and whether it is true that information about this view must be enriched with information about shadow borders.

## 4 Invisibility and Overshadows in Views

Known algorithms of visibility determination with moving viewpoint, e.g. [14, 15, 16, 17], do not resolve our problems. So, we try to resolve them below.



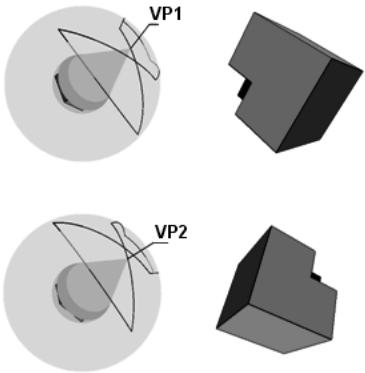
We will use two main terms: invisibility and overshadows. **Invisibility** occurs, when a face is on the opposite side of a solid (than viewpoint  $VP$ ); **overshadow** - when a face is on the same side as  $VP$ , but it is in concavity and is shaded by concavity contour.

When  $cc$  includes all edge vectors of a given face except its normal, then this face belongs to view contour, but it is inclined backwards to  $VP$ , therefore it is **invisible** and shades other faces of immersion. These shaded faces form the **area of overshadows**. **The area of overshadow is designated by viewpoint and edges of inclusion contour that belong to shading face**. When  $cc$ , during its rotation, approaches immersion, the occlusion gets smaller and finally disappears.

To resolve the problem of views with shadow we need:

1. faces of view,
2. positions of  $VP$  on the scanning trajectory at the time of visual events (change of view); two neighbour positions  $VP$  determine - for a given view, *single-view path* - segment of path (trajectory) on  $VS$  of  $VP$  motion.

So, to determine a shadow in a view it has to move along *single-view path* and test shadow displacement;



**Fig. 3.** Two different shadows on the same view

Data that needs to be collect during scanning to make view representation complete (for identification reason):

- Information about view contour.
- Information about invisible faces that lay inside view contour.
- Information about faces that are partially or completely occluded and lay inside view contour (information about border of shadow).
- *Single-view paths* for determine occlusions.

All these elements define view content. Any differences between such sets make them different

**Inclusion to immersion analysis:** Faces that are inside  $cc$  but their normals are not - belong to immersion. Faces that are visible may also belong to immersion. We use the following condition to select faces belonging immersion: normals of faces that belongs to immersion cross each other. Concavity contour is designated by border between immersion faces and other faces that belongs to view.

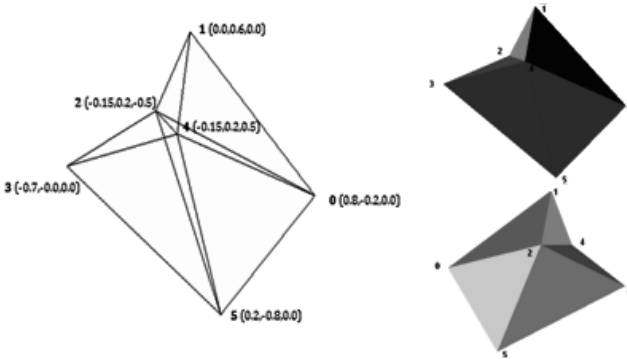
To find area of overshadows we can:

- For path  $TS_1$  connected with  $V_i$  view ( $i = 1(1)n_i$ ) of  $f_i$  face ( $i = 1(1)n$ ):
  1. First, we put straight lines from beginning point  $BP$  on scanning path  $TS$  that go through vertices of consecutive edges of concavity contour to create plane of overshadow; faces of view below this overshadow plane are occluded, faces lying above - are not
  2. Then, we step along  $TS$  path and observe changes in occlusion: note changes in view of the concavity vertex only; note them as „subviews” with determined sheets of shade;
- Finally, subviews are included into the base of views.

### 5 Summary

This latest approach to generating 3D multiview models of polyhedron (the algorithm  $MS^{UN}$ ) is in testing phase. Some monotonous polyhedron models were tested and the results are presented below. The number of views depends on angle of view cone. In presented cases, figs. 4 and 5 it was equal to 22 degrees and for this angle we obtained: 32 views for model1 and 21 views for model2, fig. 6.

Algorithm is going to be carefully checked and verified on larger set of test for monotonous polyhedrons. Expected are positive results as problem has been precisely analyzed. Result will be presented in consecutive papers.



**Fig. 4.** Model1 (concave octahedron: 8 faces, 9 edges, 6 vertices)

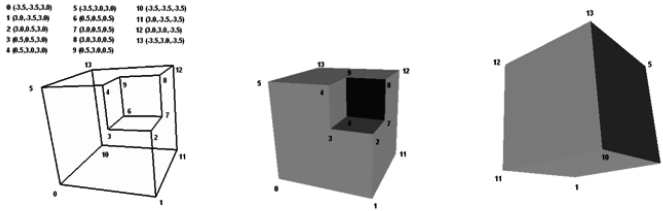


Fig. 5. Model2 (concave octahedron: 9 faces, 9 edges, 21 vertices)

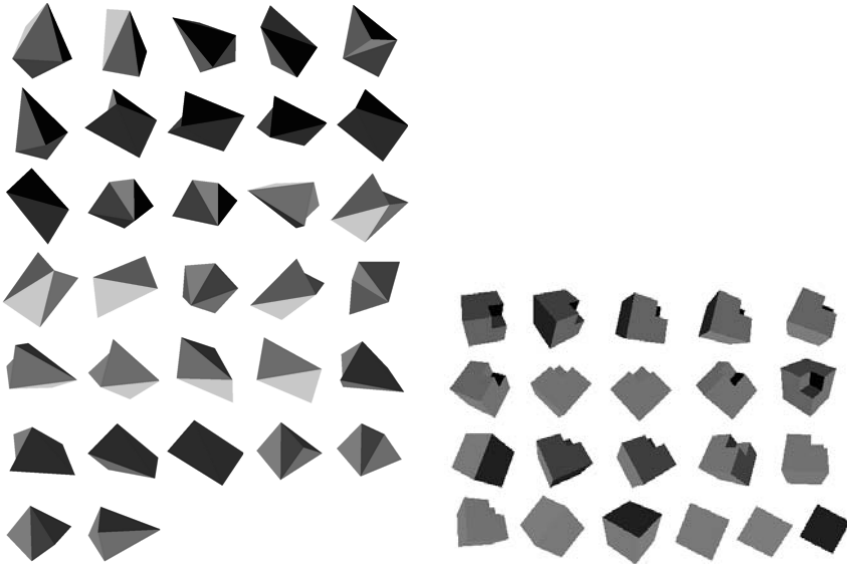


Fig. 6. a)Views for model1 b)Views for model2

## References

1. Arbel T., Ferrie F.P. 1996: Informative views and sequential recognition. Proc. ECCV'96, Cambridge, UK, April,469-481.
2. Dbkowska M., Mokrzycki W.S. 1997: Multi-view models of convex polyhedron. MGV, 6(4), 419-450.
3. Dabkowska M., Mokrzycki W.S. 1998: A new view model of convex polyhedron with feature dependent view.MGV, 7(1/2), (Proc. GKPO'98, Borki, Poland, 18-22 May), 325-334.
4. Mokrzycki W.S.: Widokowe  $2\frac{1}{2}W$  dokładne modele niewypukłych wielościanów do identyfikacji wizualnej. Raport Prace IPI PAN, nr 918, 2000.
5. Kowalczyk M., Mokrzycki W.S. 2002: A new method of finding one-view areas and tight view sphere covering. Proc. ICCVG'02,Zakopane, Poland, Sept. 25-29, 443-449.

6. Kowalczyk M., Mokrzycki W.S. 2003: Obtaining complete 2 1/2D view representation of polyhedron using concept of seedling single-view area. CV&IU 91,208-301.
7. Frydler M., Mokrzycki W.S.:  $2\frac{1}{2}D$  view models of nonconvex polyhedron on view sphere with perspective. Proc. Advance Computer Systems'03, Midzyzdroje, 22-24 X.
8. Kowalczyk M., Mokrzycki W.S.: Methods of generation 3D exact views of convex polyhedron for visual identification. Part II: Noniterative methods, implementation and tests results. MG&V, 12(4), 435-452.
9. Mokrzycki W.S.: 3W modele widokowe niewypuklych wielocianow na sferze widokowej z perspektyw. Raport Prace IPI PAN, nr 962.
10. Frydler M., Mokrzycki W.S.: New, fast algorithm of 3D multiview polyhedron representation generation on view sphere with perspective. Proc. CCV(ACV)'04, Prague, May, 15-16.
11. Kowalczyk M., Mokrzycki W.S.: Concept of 3D view representation generation of nonconvex polyhedron with view sphere partitioning on single view areas. Proc. 4th Int. Conf. on Computer Recognition Systems CORES'05, Rydzyna, May 22-25, Springer, 411-418.
12. Frydler M., Mokrzycki W.S.: Scanning faces surroundings - new concept in 3D exact multiview of nonconvex polyhedron generation. Proc. 4th Int. Conf. on Computer Recognition Systems CORES'05, Rydzyna, May 22-25, Springer, 379-386.
13. Mokrzycki W.S., Salamoczyk A.: Generating 3D multiview polyhedron representation by scanning faces surrounding. ICCVG2006, Warsaw, 25÷27 Sept.
14. Bern M., Dobkin D., Eppstein D., Grossman R.: Visibility with a moving point of view. Proc. 23<sup>th</sup> ACM Symp. on Theory of Computing, 1(3).
15. Stamos I., Allen P.K.: Interactive sensor planning. Computer Vision and Pattern Recognition. Proc. IEEE Comp. Soc. Conf., 23-25 Jun., 489-494.
16. Cohen D., Chrystanthou Y., Silva C.T., Durant F.: A survey of visibility for walkthrough applications. <http://people.csail.mit.edu/fredo/PUBLI/survey/TVCG.pdf>
17. Ghali S.: A geometric framework for computer graphics addressing modeling, visibility, and shadows. [www.cs.ualberta.ca/~ghala/thesis](http://www.cs.ualberta.ca/~ghala/thesis)

---

# On the Application of Distance Transformation in Digital Image Colorization

Przemyslaw Lagodzinski<sup>1</sup> and Bogdan Smolka<sup>2</sup>

<sup>1</sup> Silesian University of Technology, Institute of Computer Science,  
Akademicka 16, 44-100 Gliwice, Poland

[Przemyslaw.Lagodzinski@polsl.pl](mailto:Przemyslaw.Lagodzinski@polsl.pl)

<sup>2</sup> Silesian University of Technology, Institute of Automatic Control,  
Akademicka 16, 44-100 Gliwice

[Bogdan.Smolka@polsl.pl](mailto:Bogdan.Smolka@polsl.pl)

**Summary.** Colorization is a term used to describe a computerized process for adding color to black and white pictures, movies or TV programs by replacing a scalar value stored at each pixel of the gray scale image by a vector in a three dimensional color space with luminance, saturation and hue or simply RGB. Since different colors may carry the same luminance value but vary in hue and/or saturation, the problem of colorization has no inherently "correct" solution. Due to these ambiguities, human interaction usually plays a large role.

In this paper we present a novel colorization method that utilizes the morphological distance transformation and changes of neighboring pixel intensities to propagate the color within the gray scale image. The proposed method frees the user of segmenting the image, as color is provided simply by scribbles which are next automatically propagated within the image. The effectiveness of the algorithm allows the user to work interactively and to obtain the desired results promptly after providing the color scribbles. In the paper we show that the proposed method allows for high quality colorization results for still images.

## 1 Introduction

With the advancement in computer technology, adding color to gray scale images and movies in a way that looks natural to most human observers, became a problem that challenged the motion picture industry in the 1980s and has recently attracted renewed interest within the computer vision community. In the last few years, several advanced and convenient techniques for images and video have been proposed. These techniques can be grouped by the general leading idea of colorization as follows: luminance keying, color transfer [1], image analogies [2], motion estimation [3], segmentation [4], color prediction [5], probabilistic relaxation [6], chrominance blending [7].

In this paper we show that using our novel colorization method based on the new, generalized distance transformation, it is possible to obtain satisfactory colorization results in surprisingly short time and with small amount of work.

This paper is organized as follows. Section 2 presents the proposed colorization algorithm. Subsection 2.1 reviews different distance transformation methods,

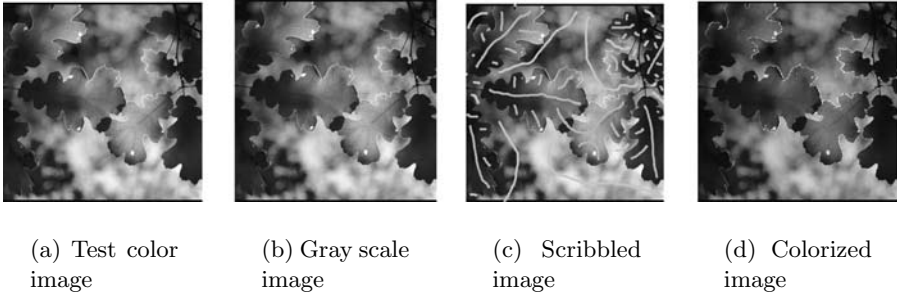
which are used to propagate the color within the gray scale image. In Subsection 2.2 we discuss the way, in which standard distance transforms can be extended to give better colorization results, and in subsection 2.3 we exploit kernel functions and introduce a correction factor for additive color mixing process in order to preserve original image structure and pixels luminance. Finally, in Section 3 we present our colorization results and in Section 4 some conclusions are drawn.

## 2 The Proposed Colorization Algorithm

In this Section we present our novel algorithm for image colorization that exploits the generalized distance transformation (DT) and luminance changes within the source gray scale image (Fig.1). Our goal is to create a fast and effective colorization algorithm which does not require precise segmentation as well as any other color images as a reference.

### 2.1 Distance Transform

The first step of our algorithm, after the user scribbles the image, is to isolate the scribbles and compute the distance from each pixel of the binary image to these scribbles. For this purpose we have implemented three different kinds of distances, [8]: 4-connectivity city-block distance ( $d_4$ ), 8-connectivity chess-board distance ( $d_8$ ) and chamfer 5-7-11 distance  $d_{ch}$ . As a result of this transformation, we obtain a gray scale image whose intensities show the distance to the object (scribble) from each image pixel.



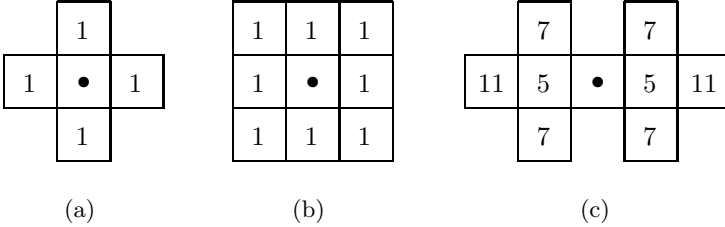
**Fig. 1.** Illustration of the proposed colorization process: (a) shows test color image that after conversion to gray scale (b) is scribbled with color (c), (d) presents the colorization result

Thus, let  $\mathcal{P}$  be a binary image defined on an image domain grid  $\mathcal{G}$  in which:

$$\langle \mathcal{P} \rangle = \{p : p \in \mathcal{G} \wedge \mathcal{P}(p) = 1\}, \quad \langle \bar{\mathcal{P}} \rangle = \{p : p \in \mathcal{G} \wedge \mathcal{P}(p) = 0\}, \quad (1)$$

are proper subsets of  $\mathcal{G}$ . For any grid metric  $d_\alpha$ , the  $d_\alpha$  distance transform of  $\mathcal{P}$  associates with every pixel  $p$  of  $\langle \mathcal{P} \rangle$  the  $d_\alpha$  distance from  $p$  to  $\langle \bar{\mathcal{P}} \rangle$ .

The  $d_4$ ,  $d_8$  or  $d_{ch}$  distance transform of  $\mathcal{P}$  are computed by scanning  $\mathcal{G}$  twice with suitable structuring element and performing a series of local operations. The structuring element consists of a pattern specified as the coordinates of a number of discrete points relative to some origin (see Fig. 2). In our algorithm the best results are obtained with chamfer 5-7-11 structuring element (Fig. 2c), since  $d_{ch}$  is the best approximation of the Euclidean distance.



**Fig. 2.** Structuring elements for:  $d_4$  (a),  $d_8$  (b) and  $d_{ch}$  (c) distance. The dot in the middle of each structuring element represents its origin.

For any  $p \in \mathcal{G}$  let  $\mathcal{B}(p)$  (before scan) be the set of pixels adjacent to  $p$  that precedes  $q$  when  $\mathcal{G}$  is scanned, and let  $\mathcal{A}(p)$  (after scan) be the remaining neighbors of  $p$ . Then, during the first scan (in left-to-right, top-to-bottom direction) we compute:

$$f_1(p) = \begin{cases} 0 & \text{if } p \in \langle \bar{\mathcal{P}} \rangle, \\ \min\{f_1(q) + 1 : q \in \mathcal{B}(p)\} & \text{if } p \in \langle \mathcal{P} \rangle. \end{cases} \quad (2)$$

After the first scan, we approach to the second scan in reverse direction i.e. right-to-left, bottom-to-top and compute the following:

$$f_2(p) = \min\{f_1(p), f_2(q) + 1 : q \in \mathcal{A}(p)\}. \quad (3)$$

Thus, after the second scan we obtain the distance values, that can be expressed as intensities of points within the gray scale image (see Fig. 3b, 3e).

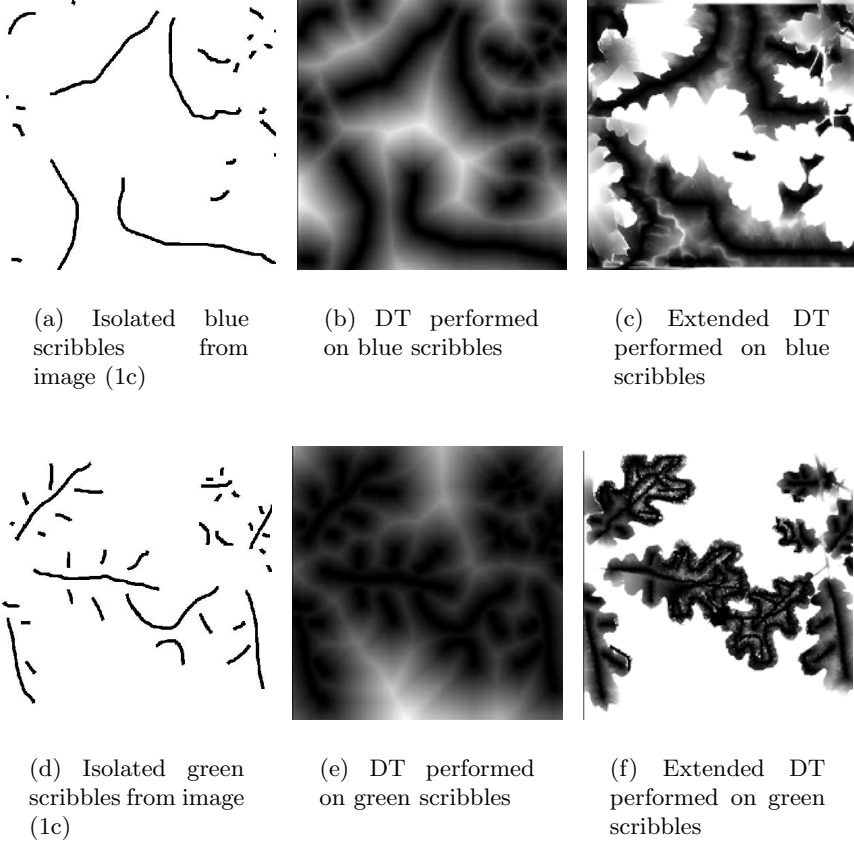
## 2.2 Extended Distance Transform

Another aspect, that is considered in our algorithm are intensity changes within the source gray scale image. We are investigating these changes in order to detect boundaries between objects and preserve the original image structures. The intensity difference between neighboring pixels is defined as:

$$\mathcal{D}(p, q) = |\mathcal{Y}(p) - \mathcal{Y}(q)|, \quad (4)$$

where  $\mathcal{Y}(p)$  denotes the intensity value at a point  $p$ .

The intensity values of neighboring points within the image are usually very close to each other and the transition between objects is quite smooth, which



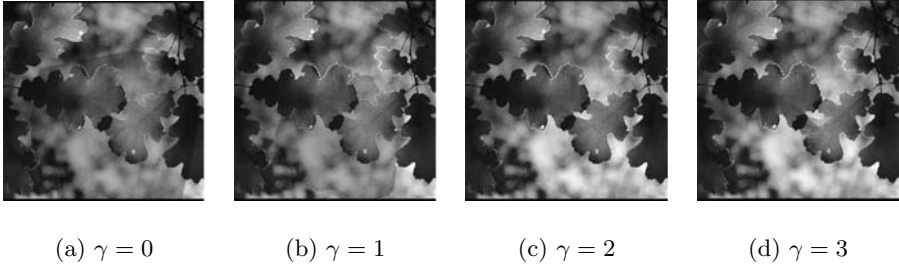
**Fig. 3.** Scribbles isolated from the image (1c) and distance transformation to these scribbles using chamfer 5-7-11 distance

causes that the boundary between them is very often hardly noticeable. In this case, in order to bring out the intensity changes, we decided to raise  $\mathcal{D}$  to the power of  $\gamma$ . Thus, we have:  $\mathcal{D}_e(p, q) = |\mathcal{Y}(p) - \mathcal{Y}(q)|^\gamma$ , where  $\gamma > 1$  denotes the exponent whose value is defined by a user. Usually, satisfactory results are obtained for  $\gamma = 2$ , (see Fig. 4).

Since the calculation of intensity changes  $\mathcal{D}_e$ , as well as computation of distances (Eqs. 2, 3), requires a sequence of local operations on neighboring pixels within the gray scale image, we have decided to merge these steps together. Taking this opportunity, we have also decided to introduce one more parameter  $\delta$ , which will let us to investigate the influence of topological distance on the quality of resulting colored images. Finally, we obtain the following equations, which define the novel, extended distance transform:

$$f_1(p) = \begin{cases} 0 & \text{if } p \in \langle \bar{\mathcal{P}} \rangle, \\ \min\{f_1(q) + \mathcal{D}_e(p, q) + 1 + \delta : q \in \mathcal{B}(p)\} & \text{if } p \in \langle \mathcal{P} \rangle, \end{cases} \quad (5)$$





**Fig. 4.** Illustration of the influence of  $\gamma$  on the quality of resulting colorization

$$f_2(p) = \min\{f_1(p), f_2(q) + \mathcal{D}_e(p, q) + 1 + \delta : q \in \mathcal{A}(p)\}, \quad (6)$$

that are next used in conjunction with gradient functions to propagate the color within the image.

In [9] two new geodesic distance transforms for gray scale images based on  $d_s$  distance which values are weighted with gray value differences were proposed. In our approach, we utilize this idea extending it with kernel functions defined in the next Section.

### 2.3 Gradient Functions

The intensities within the gray scale image can change in various ways i.e. creating a smooth transition between two shades or objects or sharp-edged boundary between objects. That suggests us the way the color, we are adding to the gray scale image, should change. To make use of these indications, we have decided to use kernel gradient functions, that are able to reproduce the structures of the source image. We have implemented two kernel gradient functions in our algorithm; linear  $f_l(d)$  and a Gaussian like  $f_g(d)$ :

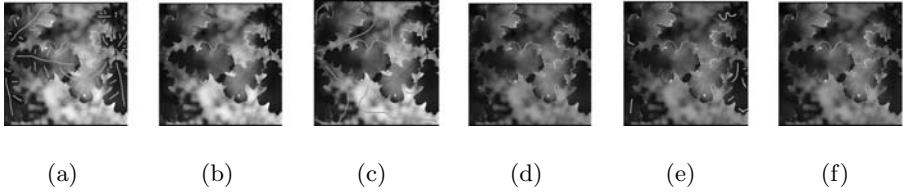
$$f_l(d) = \left| \frac{d-1}{h} - 1 \right|, \quad f_g(d) = \exp\left(- (d/h)^2\right) \quad (7)$$

where parameter  $h$ , set by the user, determines how smooth the function is. In both cases,  $d$  denotes the value we obtain from Eqs. 5 and 6, for each point of the gray scale image.

These functions are used next to determine the color  $C$  of a given point  $p$  during the additive color mixing process:

$$C(p) = C_1 \cdot f_1(d, p) + C_2 \cdot f_2(d, p), \quad (8)$$

where  $C_1$  and  $C_2$  are the colors of the scribbles inserted by the user, and  $f_1(d, p)$ ,  $f_2(d, p)$  are weights obtained for a given point  $p$  using one of the presented kernel gradient functions. Since the proposed algorithm is iterative and colors are added one by one, indexes 1 and 2 in Eqs. 8-10 correspond respectively to the current and previous colorization step, (see Fig. 5).



**Fig. 5.** Illustration of iterative colorization: images (a), (c) and (e) show color scribbled by the user, while images (b), (d) and (f) present colorization results in consecutive steps

In order to preserve the intensity from the source gray scale image in the newly propagated color, we have defined a correction factor  $r$  which allows us to determine the desired pixel intensity even when one of the RGB channels reaches the maximum value:

$$r(p) = \frac{\mathcal{Y}(p)}{\max\{C_1 \cdot f_1(d, p), C_2 \cdot f_2(d, p)\}} \quad (9)$$

where  $\mathcal{Y}(p)$  denotes the original intensity value from the source gray scale image. Thus, the final color can be achieved from:

$$C(p) = [C_1 \cdot f_1(d, p) + C_2 \cdot f_2(d, p)] \cdot r(p), \quad (10)$$

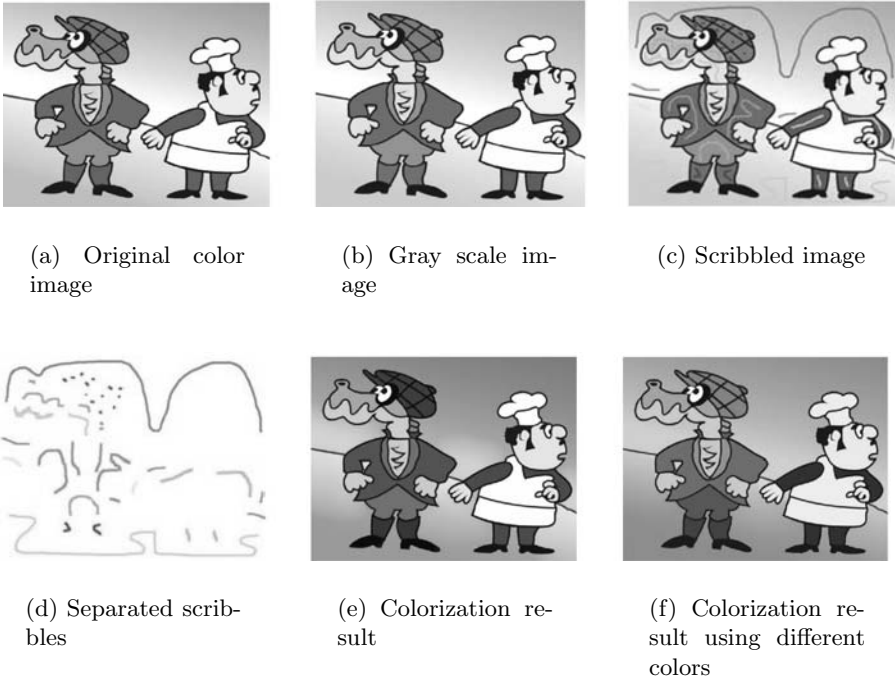
In this way, we are able to determine the color for each point of the image taking the colors scribbled by a user as a basis, and propagate them, preserving the original intensity values within the color image. Our method uses the *max* function within the RGB color space to determine the luminance value. However, the proposed method can exploit various fuzzy norms, other color spaces like HSV or  $\alpha\beta$  and various methods of luminance calculation within the color image.

### 3 Colorization Results

The results shown here were all obtained using the presented method that works on the basis of the extended distance transformation. The proposed solution is iterative and adds scribbled colors one by one. This method was implemented using Microsoft Visual C# .NET 2.0. Although the code is not yet fully optimized, this implementation of algorithm works fast enough to allow the user for interactive work without noticeable delays and achieving real-time preview.

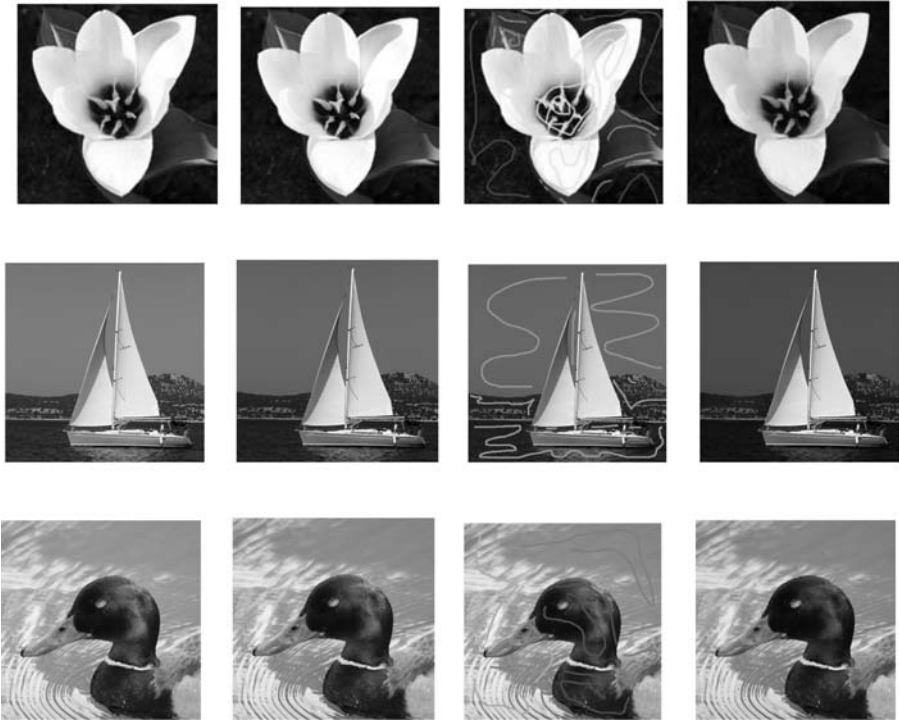
The colorization results look very promising. The proposed method gives very good results for images with solid outlines like cartoon images (Fig. 6) as well as for natural images (Fig. 7). The proposed scheme can be successfully used for natural images colorization and to enhance scientific presentations or medical illustrations.

In Fig. 6 we can see the cartoon colorization example. After conversion of the original color image (Fig. 6a) to the gray scale (Fig. 6b) a user scribbled the image with a chosen color (Fig. 6c). The scribbles are next isolated from the image (Fig. 6d) converted to the binary form and then our colorization method is applied. As a result we obtained a good quality color image (Fig. 6e), which looks very similar to the source image (Fig. 6a). It is also possible to provide a different set of colors and propagate them within the image. Such example is shown in the Fig. 6f.



**Fig. 6.** Cartoon colorization example: image from Polish cartoon "Przygody Baltazara Gabki"

The presented method can be also successfully applied to natural images. In Fig. 7 we present a set of natural images colorized using our algorithm. In columns from left to right we can see original color images as well as images converted to gray scale, the ones scribbled with color and results of our colorization. Comparing original color images with these obtained during colorization process we can see some differences which result in most cases from the choice of colors used for the scribbles. Apart from this effect, our colorization examples look very natural, transitions between colors are smooth and edges are sharp. Thus, the structure, look and feel of the original image can be preserved.



**Fig. 7.** Colorization examples obtained with the proposed algorithm. In columns: original color images, original gray scale images, scribbled images and our colorization results.

## 4 Conclusions

In the future work, we plan to explore other than RGB color spaces i.e. *CIE Lab* or  $l\alpha\beta$  color spaces, which minimize the correlation between channels and give direct access to the luminance channel. Additionally, we will experiment with other distance transformation algorithms and fuzzy weighting functions in order to make this method more sensitive to the hardly detectable boundaries. The optimization of the code of our application will allow us also to shorten the time of colorization process in the case of very large images. We are very optimistic that these improvements will allow us to produce even better colorization results in the near future.

## References

1. E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications* **21**, pp. 34-41, May 2001.
2. T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to grayscale images," in *Proceedings ACM SIGGRAPH Conference*, pp. 277-280, 2002.

3. Z. Pan, Z. Dong, and M. Zang, "A new algorithm for adding color to video or animation clips," in *Proceedings WSCG, International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 515-519, 2004.
4. A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proceedings ACM SIGGRAPH Conference*, pp. 689-694, 2004.
5. J. S. Madeira, A. Stork, and M. H. Grob, "An approach to computer-supported cartooning," *The Visual Computer* **12**, pp. 1-17, 1996.
6. T. Horiuchi, "Colorization algorithm for gray-level images by probabilistic relaxation," in *Proceedings IEEE International Conference on Pattern Recognition*, pp. 867-870, 2003.
7. L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending", *IEEE Transactions on Image Processing* **15**, pp. 1120-1129, May 2006.
8. R. Klette and A. Rosenfeld, *Digital Geometry - Geometric Methods for Digital Picture Analysis*, Morgan Kauffman, 2004.
9. P. J. Toivanen, "New geodesic distance transforms for gray-scale images," *Elsevier - Pattern Recognition Letters* **17**, pp. 437-450, 1995.

---

# Local Analysis of Stereo Image Pairs with Polynomial Series

Przemysław Głomb

Institute of Theoretical and Applied Informatics, Bałtycka 5, 44-100 Gliwice, Poland  
przemg@iitis.gliwice.pl

**Summary.** The focus of this article is the analysis of corresponding pixel neighborhoods in two stereo pair images. A nonlinear structure present in the form of multiplicative pixel interactions is investigated with polynomial Volterra predicting model. The difference in efficiency of central pixel prediction on subsequent images is the base for quantifying uncertainty introduced with second camera offset. The main contribution of this article are experimental results and the discussion of applying implicit polynomial models based on reproducing kernel regression to the Amsterdam Library of Object Images database stereo image set.

## 1 Introduction

Humans unwittingly use stereo vision for getting around in three dimensional world. The brain is very effective in matching and extracting information from a pair of stereo images. At the same time, processing of such pairs is also the key to many advanced computer vision applications. Local region processing forms the backbone of a substantial number of such methods [1]. On the other hand, it is known [7] that commonly used second order metrics, like normalized cross-correlation (NCC) are blind to the structure present in higher order statistics (HOS). This structure contains an important part of information about scene semantics, yet is difficult to extract because computational load required for its effective estimation [6].

One way to study this structure is by the analysis of predictability: how much can we tell about one region in first stereo image from the corresponding other region? The efficiency of such prediction for subsequent images can be compared and described. Certain models, including used in this work Volterra polynomial filters, are directly related to HOS of the region, allowing for extraction of statistics that can be later used in construction of matching methods. Recently a new method for effective general identification of arbitrary order polynomial model has been introduced [3]. In this article, we follow their earlier work [4], applying that approach to stereo images.

We present results and analysis of experiments in pixel value prediction from a multiplicative interactions of neighborhood values. We investigate polynomial models of various order and different neighborhood sizes. The data set used is the stereo pair set from Amsterdam Library of Object Images [5].

The paper is structured as follows: second section presents the method used for analysis, third section presents the experiment results, last section presents conclusions.

## 2 Method

We focus on a local pixel neighborhood of predefined size (see figure 1). We use a polynomial model of multiplicative pixel interactions [4] to predict the central value and corresponding pixel value in the other (pair) image. We then compare the difference in prediction efficiency in the function of polynomial order and neighborhood size.

We are given two calibrated stereo images  $\mathcal{Y}_1$  and  $\mathcal{Y}_2$ , both indexed by  $\mathbf{n} = (n_1, n_2)^\top$ . For some fixed  $\mathbf{n}$ , we get two central points  $y_1 = \mathcal{Y}_1[\mathbf{n}]$  and  $y_2 = \mathcal{Y}_2[\mathbf{n}]$ , and the neighborhood vector<sup>1</sup>  $\mathbf{x} = (x_1, x_2, \dots, x_N)$ , where  $x_i = \mathcal{Y}_1[\mathbf{n} + \boldsymbol{\delta}_i]$ . The set of offsets  $\{\boldsymbol{\delta}_i\}_{i=1}^N$  is used to define the neighborhood (which does not include the  $y_1$ ).

We model the dependency of both central pixels on the neighborhood  $\mathbf{x}$  as an  $L$ -th order polynomial Volterra system [6]

$$y_i(\mathbf{x}) = H_0^i + H_1^i \mathbf{x} + H_2^i \mathbf{x} + \dots + H_L^i \mathbf{x} \quad (1)$$

where

$$H_0^i = h_0^i, \quad H_j^i \mathbf{x} = \sum_{n_1=1}^N \sum_{n_2=1}^N \dots \sum_{n_j=1}^N h^i(n_1, n_2, \dots, n_j) x_{n_1} x_{n_2} \dots x_{n_j}. \quad (2)$$

To identify above model, [3] use the linear regression in Reproducing Kernel Hilbert Space (RKHS), where they find function  $y_i(\mathbf{x})$  as

$$y_i(\mathbf{x}) = \sum_{j=1}^M \gamma_{ij} k(\mathbf{x}, \mathbf{z}_j), \quad k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=0}^L \left( \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right)^l, \quad (3)$$

for given model order  $L$ , a set of training neighborhoods  $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^M$  to arrive at parameters  $\gamma_{ij} \in \mathbb{R}$ .

Given set  $\mathcal{X}$  and order  $L$ , the value of the function can be calculated [3] from the expression

$$y_i(\mathbf{x}) = \mathbf{z}_i^\top K^{-1} \mathbf{k}(\mathbf{x}), \quad (4)$$

where  $\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), \dots, k(\mathbf{x}, \mathbf{x}_M))^\top \in \mathbb{R}^M$ ,  $K$  is a Gram matrix defined as  $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ , and  $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{iM})^\top$ ,  $z_{ij}$  is the central value in image  $i$  associated with, but not belonging to, training neighborhood  $\mathbf{x}_j$ .

Following [4], we extracted the set  $\mathcal{X}$  from the database, then used (4) to observe the error  $|z_i - y_i(\mathbf{x}_i)|$ . We thus didn't separate the data into training

---

<sup>1</sup> For simplicity, we don't consider here image boundaries.

and test sets, as the objective was to investigate the structure of images, not generalization over to other stereo pairs.

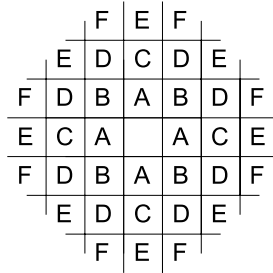
We note that we can recover (1) from (4) (i.e. using the method from [2]). The original Volterra filter can be then represented as a dot product of vector of stacked coefficients  $\mathbf{h}_i$  and vector  $\phi(\mathbf{x})$  of polynomial combinations of components of  $\mathbf{x}$ ,  $y_i(\mathbf{x}) = \mathbf{h}_i^T \phi(\mathbf{x})$ . Since the stacked coefficient vector  $\mathbf{h}_i$  is related to the autocorrelation matrix  $R_{xx}$  of members of  $\mathcal{X}$  and cross correlation  $\mathbf{p}_i$  of components of  $\mathbf{z}_i$  with members of  $\mathcal{X}$  with the relation [6]  $\mathbf{h}_i = R_{xx}^{-1} \mathbf{p}_i$ , we see that used model directly relates to HOS of region components.

To quantify the prediction error, we use the Peak Signal to Noise Ratio (PSNR) distance, defined as

$$\text{PSNR}(y_i) = 10 \log_{10} \frac{255^2}{\frac{1}{M} \sum_{j=1}^M |z_i - y_i(\mathbf{x}_i)|^2} \tag{5}$$

### 3 Experiments

We performed experiments with the ALOI database. Among others, that image set contains 750 sequences wide-baseline stereo images of different physical objects. Each object has three images associated with it: base image, camera rotated  $15^0$  and  $30^0$ . Each object is in the center of the camera view<sup>2</sup>. Several combinations of stereo pairs can be constructed; we’ve used all three images for three independently identified polynomial models.



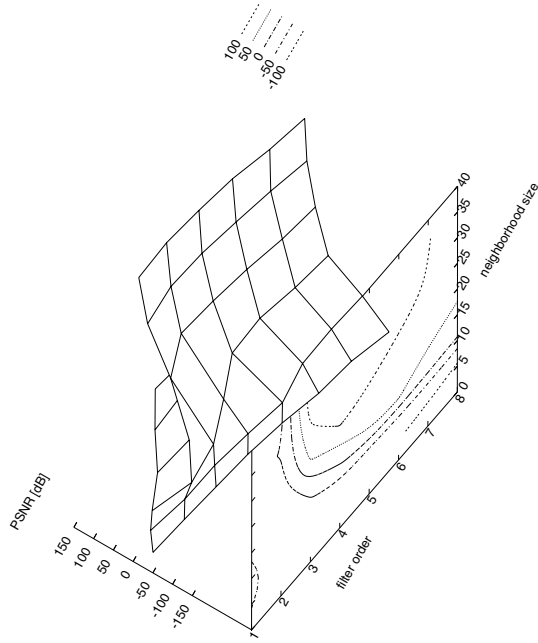
**Fig. 1.** Various neighborhoods used in experiments. The subsequent neighbors from one class and all classes below are used together. Note that the central pixel is not included.

We use various neighborhood sizes, presented on figure 1. Each next neighborhood is used together with previous, so we predict from sets  $(A)$ ,  $(A, B)$ ,  $(A, B, C)$ , etc. The regression was done with C++ implementation using Matlab numerical routines. Model order varied from 1 to 8.

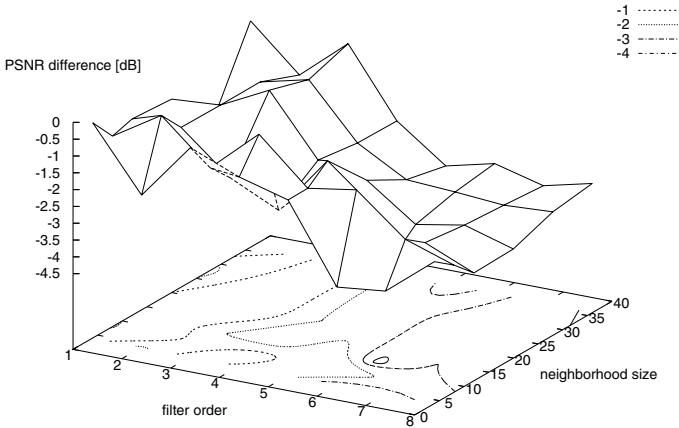
---

<sup>2</sup> This is important as it justifies the application of the above method.





**Fig. 2.** PSNR of prediction performance for central pixel from it's neighborhood in function of its size and model order. Negative values are associated with numerical errors of matrix inversion. See text.



**Fig. 3.** PSNR difference for performance of prediction of central pixel in base and the other stereo image, for  $15^0$  camera rotation variant. Note the overfitting and thus reduced performance of higher orders with larger neighborhood.

Figure 2 presents the base image prediction result in function of order and neighbor size. Negative values of PSNR are associated with numerical errors introduced by ill-conditioning of Gram matrix. Regularization methods or pseudo-inverse could possibly improve results in those regions. We observe that higher order requires smaller neighborhood to achieve some fixed performance than lower order. For some neighborhood size, the order value that produces maximum efficiency can be usually observed, but since error is negligible over a range of orders, this is only important for small neighborhoods. Prediction performance grows with each new set of neighborhood points added, this relationship is non-linear.

Figure 3 presents the difference in PSNR for prediction of  $15^\circ$  stereo pairs. Lower order results are again contaminated with numerical noise. The better performance of lower order models than higher ones is due to better generalization properties of the former and overfitting of the latter. Interestingly, this is less pronounced for  $30^\circ$  (see figure 4). We don't note the increase of performance with neighborhood size growth. On overall, the error level change is small. The uncertainty introduced with moving the second camera are on the level of those contributed by individual pixels.

We've also observed the behavior of errors. With different values of PSNR, the error histogram has similar shape, with strong peak. This suggests that most of the predictions are performed correctly, with only few errors of large magnitude. Evolution of error kurtosis with change of parameters is presented on figure 5. The error streams are highly correlated, with correlation coefficient on the order of  $r = 0.9$ .

It is interesting to observe the computational cost of prediction realization. The method introduced in [3] has the property of having complexity dependent linearly on number of  $\mathcal{X}$  samples and neighborhood size. Depending on value of used parameters, this can be several order of magnitude more or less than direct

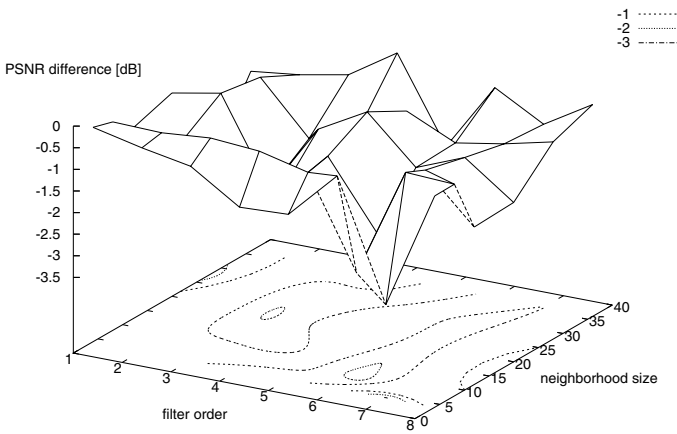
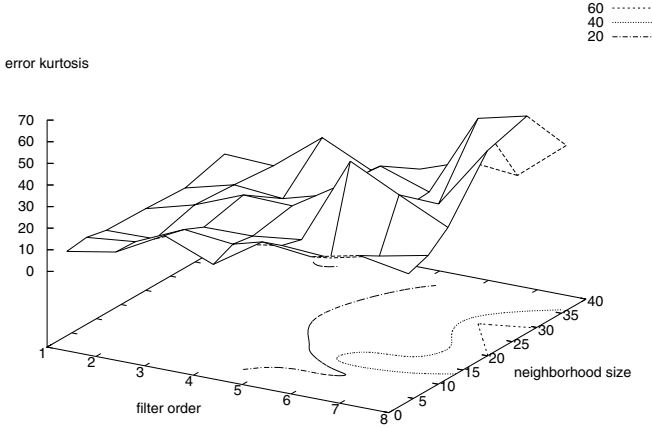
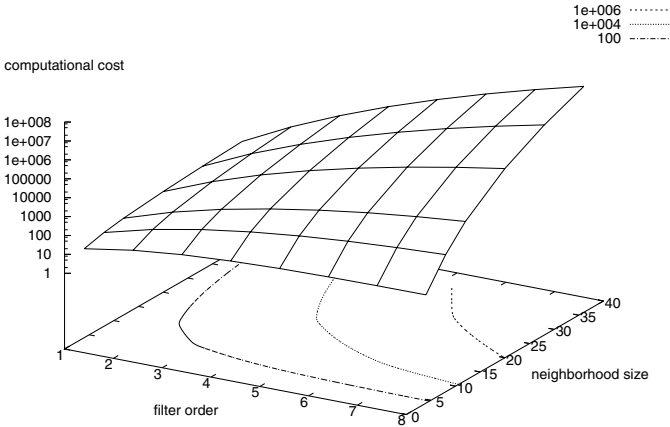


Fig. 4. Similar results of PSNR difference for the  $30^\circ$  camera rotation variant



**Fig. 5.** Kurtosis of the prediction errors for base image central pixel prediction



**Fig. 6.** Computational cost associated with direct realization of symmetrized Volterra filter in the form of (1)

realization of (1). The latter cost grows quickly with small changes in parameter values (see figure 6).

## 4 Conclusion

We have presented the results of application of implicit polynomial models for prediction and structure representation of local neighborhood in stereo pair images. The implicit Volterra models were identified for varying neighborhood sizes and model orders. Prediction performance was compared against parameters and stereo disparity value.

Although preliminary, these results confirm that significant higher order structure is present in local neighborhood. It is not clear how to effectively include that information for image matching, but presented results can be used to propose a range of orders and neighborhood sizes for methods working on local image region. Application of results could include, but is not limited to: parameter setting for methods for matching, compression (i.e. probability estimation for arithmetic coding), and shape from stereo. At the same time, analysis of results suggest that more research should be done on higher order relations between corresponding pixel neighborhoods.

**Acknowledgement.** This work was partially supported by by Polish Ministry of Science and Higher Education project 3T11C02827.

## References

1. M.Z. Brown, D. Burschka, and G.D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.
2. T.J. Dodd and R.F. Harrison. Estimating volterra filters in Hilbert space. In *Proc. of the IFAC Conference on Intelligent Control Systems and Signal Processing (ICONS)*, pages 538–543, 2003.
3. M. O. Franz and B. Schlkopf. A unifying view of Wiener and Volterra theory and polynomial kernel regression. *Neural Computation*, 18(12):3097–3118, 2006.
4. M.O. Franz and B. Schlkopf. Implicit wiener series for higher-order image analysis. In *Advances in Neural Information Processing Systems*, volume 17, pages 465–472. MIT Press, 2005.
5. J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.
6. J. Mathews and G.L. Sicuranza. *Polynomial Signal Processing*. Wiley, 2000.
7. G. Sicuranza and S. Mitra, editors. *Nonlinear Image Processing*. Academic Press, 2000.

---

# The Architecture of the Face and Eyes Detection System Based on Cascade Classifiers

Andrzej Kasiński and Adam Schmidt

Institute of Control and Information Engineering str. Piotrowo 3a 60-965 Poznan,  
Poland

Andrzej.Kasinski@put.poznan.pl, Schmidt.Adam@gmail.com

**Summary.** The precise face and eyes detection is crucial in many Human-Machine Interface system. The important issue is the reliable object detection method. In this paper we present the architecture of a 3-stage face and eye detection system based on the Haar Cascade Classifiers. By applying the proposed system to the set of 10000 test images the 94% of the eyes were properly detected and precisely localized.

## 1 Introduction

Many Human-Machine Interface (HMI) tasks, such as face tracking, expression recognition and human person recognition to be efficient require the proper initialization. For example, face recognition (FR) techniques are sensitive to the accurate face alignment. Detecting only a face is often insufficient to achieve the desired final classification results. Information such as face in-plane rotation, scale and precise location can be obtained by localizing eyes on previously extracted faces.

As precise eyes location enables accurate alignment, one has to first design an efficient face and eye localization method in order to develop an automatic face recognition system.

Currently, the Haar Cascade Classifiers (HCC) are getting increasing attention. High detection ratio obtained with those computationally-efficient detectors suggests the possibility of using them in a reliable real-time HMI systems. Therefore, our goal was first to train the efficient face and eyes HCC detectors and then to combine them into a hierarchical system. Moreover, we wanted to improve the detection rates of the HCC by introducing the additional knowledge-based criteria.

In this paper we present a 3-stage hierarchical face and eye detection system based on the HCC. Firstly, we present the state of art in both face and eyes detection. Secondly, we present the idea of the HCC. Then we describe the architecture of our system. Finally, give the preliminary results and the achieved detection rates.

## 2 The State of Art in Face and Eyes Detection

A human face is a highly non-rigid 3D object whose image is susceptible to both pose and expression variations. This combined with variability of personal face

features and possible structural disturbances (such as glasses, facial hair, make-up) makes face detection a challenging task. The poor performance of features detectors based only on human knowledge as well as of those based on simple template matching, points out to the necessity of involving machine learning-based approaches.

Huang et al. [1] applied the Polynomial Neural Network (PNN) to the task of detecting faces. PNN is a single-layer network taking polynomial expansion of pattern features as inputs. The feature pool was based on pixel intensity values, Sobel filter responses, and on directional gradient decomposition. The Principal Components Analysis (PCA) was then used to reduce feature's vector dimension. It was proved that the system based on gradient decomposition outperformed systems using simpler features.

The system proposed by Heisele et al. [2] consisted of three independent first-level Support-Vector Machine (SVM) detectors for finding potential eyes, nose and mouth regions. The second-level classifier checked if their relative position could correspond to that typical for the human face. Bileschi and Heisele [3] improved this method by training SVM not against a rich background but only against some other facial features. The authors claimed that component-based face detection system is more resistant to face pose changes than holistic detectors.

Viola and Jones [4] were first to introduce Haar Cascade Classifiers and to apply HCC to the task of face detection. The idea of using cascade of simple classifiers led to the creation of an accurate and computationally efficient detection system. Lienhart et al. [5] improved the HCC by enlarging the feature pool with the rotated Haar-like features. He also tested the influence of various weak classifiers and boosting algorithms on the performance of cascades.

Weak classifiers ensembles were also used by Meynet et al. [6]. They combined a simple HCC with another parallel weak classifiers ensemble. The HCC was used to discard easy to classify non-faces. The remaining windows have been tested with boosted classifiers based on the Anisotropic Gaussian Features; the final classification depending on votes of those classifiers.

Eyes detectors are usually applied to already localized face regions, which are fairly similar. Here the detectors must discriminate between eyes and other facial features.

Wang et al. [7] used homomorphic filtering to compensate for illumination variations. The binary template matching was then applied to preprocessed images in order to extract potential eyes. Candidate regions were verified with the SVM and the precise eyes' location was acquired with variance filters.

Many authors tried to use the HCC in the task of eyes detection. Wilson and Fernandez [8] used cascades trained against other features to extract eyes, a mouth and a nose from the face region. As the processing of a whole face led to many false positives (FP) they proposed the regionalized search approach. This explicitly means the use of the knowledge about a face structure i.e. looking for a left eye in an upper-left, for a right eye in an upper-right, a nose in a central and a mouth in a lower part of the face. This simple solution significantly reduced the FP ratio.

Feng et al.[9] used the HCC at the first stage of their detection system. In order to reduce the FP ratio the results have been verified with another boosted classifier, the one based on ordinal features rather than on Haar-like and trained with the algorithm similar to the AdaBoost.

### 3 The Haar Cascade Classifiers

The HCC emerged as a successful combination of three ideas. Instead of working directly on image function values, the detector uses an extensive set of features, which can be efficiently computed in a fixed time. This feature-based approach helps to reduce the in-class variability and increases variability between classes. Secondly, using a boosting algorithm allows for a concurrent selecting of a small subset of sufficient features and for the classifier training. Finally, creating a cascade structure of gradually more complex classifiers results in a fast and efficient detection scheme.

#### Haar-Like Features

According to Lienhart [5], Haar-like features can be calculated with the following equation:

$$feature = \sum_{i \in \{1 \dots N\}} \omega_i \cdot RecSum(x, y, w, h) \quad (1)$$

Where  $RecSum(x, y, w, h, \phi)$  is the sum of intensity values over any given upright or rotated rectangle enclosed in a detection window and  $x, y, w, h, \phi$  stand for coordinates, dimensions and rotation of that rectangle (see Figure 1).

To reduce potentially infinite number of features, the following restrictions are applied:

- Pixel sums over only two rectangles are allowed ( $N=2$ )
- The weights are used to compensate the area difference of two rectangles and have opposite signs. Which means that  $\omega_0 \cdot Area(r_0) = -\omega_1 \cdot Area(r_1)$ , substituting  $\omega_1 = 1$  one gets  $\omega_0 = -Area(r_0)/Area(r_1)$
- The features should be similar to those used in early stages of human vision pathway, such as directional responses of Gabor filters.

Those constraints leave 14 prototype features (see Figure 2), which can be scaled in both directions and placed in any part of the detection window. This allows to create an extensive feature pool. The features are computed as the proportion of pixel sums under black and white rectangles and scaled to compensate for the areas difference. It's worth mentioning, that line features can also be viewed as a combination of two rectangles: one of them containing both black and white, but the second only a black area.

To efficiently evaluate features, two novel image representations are introduced. The Summed Area Table ( $SAT(x, y)$ )[4] is used to calculate features based on upright rectangles. Each entry of the table is defined as the sum of

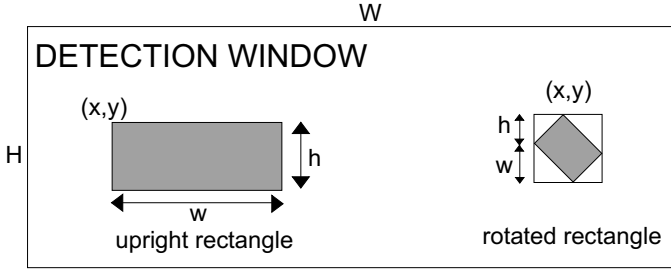


Fig. 1. Upright and rotated rectangles in the detection window

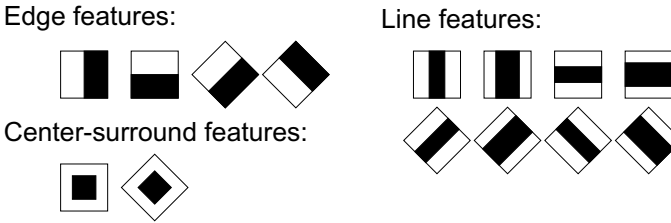


Fig. 2. Prototypes of Haar-like features

pixel intensities over the upright rectangle extending from (0, 0) to (x, y) and is being filled according to the formula:

$$SAT(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \tag{2}$$

Once filled, the SAT enables computing pixel sum over any upright rectangle with only four look-ups:

$$\begin{aligned} RecSum(x, y, w, h, 0) = & SAT(x - 1, y - 1) + \\ & + SAT(x + w - 1, y + h - 1) - SAT(x + w - 1, y - 1) - \\ & - SAT(x - 1, y + h - 1) \end{aligned} \tag{3}$$

Rotated features are computed using another auxiliary representation called the Rotated Summed Area Table [5]. Each entry is filled with the following value:

$$RSAT(x, y) = \sum_{|x-x'| \leq y-y', y' \leq y} I(x', y') \tag{4}$$

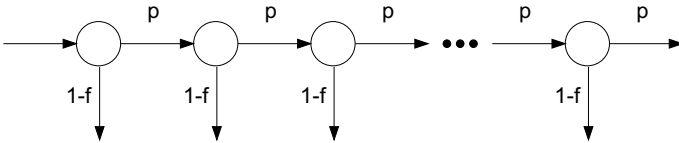
Pixel sum of any rotated rectangle can be computed according to:

$$\begin{aligned} RecSum(x, y, w, h, 45) = & RSAT(x - h + w, y + w + h - 1) + \\ & + RSAT(x, y - 1) - RSAT(h - x, y + h - 1) - \\ & - RSAT(x + w - 1, y + w - 1) \end{aligned} \tag{5}$$



### Classifiers Cascade

In the most of cases the detected object occupies only a small part of the image. Thus it's better to discard non-object regions quickly and focus only on those which are relevant, than to examine every window thoroughly. The cascade structure allows for such an approach. It consists of the  $N$  stages i.e. of serially connected classifiers distinguishing between the detected object and the background. Every stage is trained to achieve TP ratio  $p$  close to 1 with the FP ratio  $f$  kept usually 0.5. The positively classified windows are passed to the subsequent stage; the others are excluded from the further processing.



**Fig. 3.** Structure of the cascade detector

Due to the serial nature, the overall detection ratios are exponential function of single stage efficiencies:

$$TP_{cas} = \prod_{i=1}^{i \leq N} p_i \approx p^N \quad (6)$$

$$FP_{cas} = \prod_{i=1}^{i \leq N} f_i \approx f^N \quad (7)$$

The adequate selection of  $p$ ,  $f$  and  $N$  results in a detector achieving a high TP ratio (slightly less than 100%) and a low FP ratio at the same time. The stages are consecutively trained to achieve the desired detection rates. The whole sets of positive and negative samples are presented only to the first stage classifier. The others are trained only with subsets which have passed previous stages. In that way classifiers at successive stages are faced with more difficult tasks and have to discover subtler differences to keep up the desired  $p$  and  $f$  ratios.

### The Single Stage Classifier

Having such extensive features pool one needs to find a way to select a minimal subset guaranteeing the desired detection rate. Boosting is a machine learning concept combining performance of many 'weak classifiers' (where 'weak' means only slightly better than a random choice) into a single powerful ensemble called 'strong classifier'.

In the HCC simple CARTs (Classification And Regression Tree) are used as weak classifiers. Their size is restricted only to several splits. In the simplest

case (single-split CARTs called "stumps") they rely on a single feature only. Using slightly more complex weak classifiers (e.g. 4-splits CARTs) slows down the training but allows for preserving some relations between features in a weak classifier. Even those more complex classifiers could not be sufficient to achieve the desired detection rates. To assemble weak classifiers into a strong one the boosting algorithm called AdaBoost[10] is used. In [5] Lienhart with colleagues proved that by using the version called the Gentle AdaBoost one obtains a detector having lower FP ratio than those detectors trained by using other AdaBoost versions.

*Gentle Adaboost algorithm specification according to [10]:*

1. Given  $N$  examples  $(x_1, y_1), \dots, (x_N, y_N)$  where  $x_i \in R^k, y_i \in (-1, 1)$
2. Start with weights  $w_i = 1/N, i = 1, \dots, N$
3. Repeat until  $p$  and  $f$  are achieved, for  $m = 1, \dots, M$ 
  - a. Fit the regression function (the CART)  $f_m(x)$  by the weighted least-squares of  $y_i$  to  $x_i$  having weights  $w_i$
  - b. Set  $w_i = w_i \cdot \exp(-y_i \cdot f_m(x_i))$
4. Output the classifier:  $sign[\sum_{m=1}^M f_m(x)]$

## 4 The System's Architecture

Our detection system consists of the three stages. At the first one, the HCC face detector is applied to the whole image. As the neighboring positive responses of HCC are merged into a single detection result, it is possible to fine-tune the detection ratios by relaxing the constraint on the minimum number of neighbors ( $Nb$ ). The detected candidate regions are further processed independently.

At the second stage, the left and right eye HCC detectors are used on previously found face regions and their results are stored in the two lists. As with the face detector the constraint on the minimum number of merged neighbors can be set. Moreover, instead of searching for eyes over the whole face, the regionalized search can be used. This means, applying the left eye detector the rightmost and the right eye detector to the leftmost 60% of the upper half of the face.

The third stage is a simple knowledge-based rule of combining left and right eye detections into valid eye-pairs. For each left and right eye combination in a given face rectangle an in-plane rotation  $\phi$  is calculated. Eye-pairs with  $\phi > 20$  are discarded, as too unlikely to belong to a upright view of the face.

## 5 The Results and Conclusions

Our HCCs were trained with the Gentle AdaBoost by using 4-split CART as a weak classifier and setting the required TP ratio of each stage for 0.999.

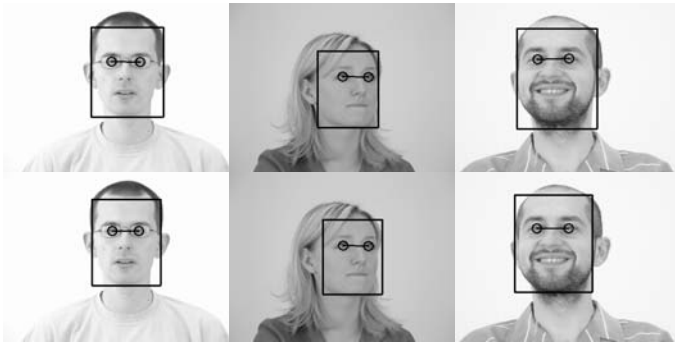
The positive training set for the face HCC consisted of 2500 face pictures from our base. The negative set was created by randomly gathering 3500 images not containing any faces. Eyes detectors were trained with positive sets of 2500 left or right eye images from our base. Negative sets consisted of face images with the appropriate eye being hidden.

The eye localization error measure was the same as the one used by Campadelli [11].

$$error = \frac{\max(\|C_l - C_{lGT}\|, \|C_r - C_{rGT}\|)}{\|C_{lGT} - C_{rGT}\|} \quad (8)$$

Where  $C_l$  stands for the center of the left eye found,  $C_r$  stands for the center of the right eye found,  $C_{lGT}$  and  $C_{rGT}$  stand for the centers of ground truth eyes.

Detections with error less than 0.1 were treated as TPs, the others were counted as FPs. Pictures without any positive eyes detection were considered to be false negatives (FNs).



**Fig. 4.** Exemplary results: upper row - manually marked face and eyes, lower row - automatically detected face and eyes

While no constraints to the minimum number of neighbors were set, the proposed system achieved the TP ratio of 94% and FP ratio of 13%. Constraining solely the eyes detector ( $Nb=3$ ) resulted in the TP ratio of 88% with less than 1% FP. The average processing time on a PC with Intel Celeron 2,8 GHz processor and 512 MB RAM was 321 ms.

Our detection system was proved to be efficient both w.r.t detection rates and computation costs. It turned out to be resistant to pose variations and to structural disturbances.

Future work will focus on using this solution in a complete face recognition system. The influence of the training sets used, of the weak classifiers applied and of the each stage's desired detection rates settings on the overall detection ratios will be investigated. Additionally, our detectors performance will be compared with the performance of custom HCCs trained by other authors.

## References

1. Huang L-L, Shimizu A, Hagihara Y and Kobatake H (2003) Gradient feature extraction for classification-based face detection. *Pattern Recognition* 36:2501:2511
2. Heisele B, Serre T, Pontil M and Poggio T (2001) Component-based Face Detection. *CVPR* 01:657–662
3. Bileschi S, Heisele B (2002) Advances in Component-Based Face Detection. In: *Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*. Springer-Verlag, London, UK
4. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: *Proceedings of CVPR* 1:511–518
5. Lienhart R, Kuranov A, Pisarevsky V (2002) Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. Technical report, Microprocessor Research Lab, Intel Labs
6. Meynet J, Popovici V, Thiran J (2005) Face Detection with Mixtures of Boosted Discriminant Features. Technical report, EPFL
7. Wang Q, Yang J (2006) Eye Detection in Facial Images with Unconstrained Background. *Journal of Pattern Recognition Research*, 1:55–62
8. Wilson P, Fernandez J (2006) Facial feature detection using Haar classifiers. *J. Comput. Small Coll.* 21:127–133
9. Feng X, Wang Y, Li B (2006) A fast eye location method using ordinal features. In: *Proceedings of the 2006 ACM SIGCHI*
10. Freund Y, Schapire R (1996) Experiments with a New Boosting Algorithm. In: *Proc. of International Conference on Machine Learning*
11. Campadelli P, Lanzarotti R, Lipori G (2006 (preprint)) Eye localization: a survey. In: *The fundamentals of verbal and non verbal communication and the biometrical issues*. NATO Science Series

---

# Morphological Edge Detection Algorithm and Its Hardware Implementation

Marek Kraft and Andrzej Kasiński

Institute of Control and Information Engineering, Poznań University of Technology,  
Piotrowo 3A, 60-965 Poznań  
Marek.Kraft@put.poznan.pl

**Summary.** Mathematical morphology is a widely known image processing technique. Morphology-based processing techniques can be easily implemented, thanks to conceptual simplicity of basic morphological operators. However, most morphological algorithms that are run on sequential processors are unable to meet real-time requirements. A way to solve this problem are the implementations using dedicated hardware, e.g. the FPGAs.

This paper describes a complete FPGA architecture for real-time edge detection. The architecture consists of prefiltering, gradient image computation and non-maximum suppression blocks, allowing the extraction of a single-pixel-thin binary edge image of good quality. The described solution, although simple and compact, is capable of processing image data stream with the speed of hundreds of frames per second for a  $512 \times 512$ , 8-bit gray-scale image, while keeping the costs, size and power consumption at a low level. Description of implemented algorithm, resource usage summary, output images, as well as block diagrams of the architecture are included in the paper.

## 1 Introduction

Edge detection refers to the process of identifying and locating significant discontinuities in an image function, to extract significant external world information – the scene structure. Edge detection is an important step in the feature extraction and supervision tasks, like texture analysis, pattern recognition, machine vision, motion detection etc. Therefore, fast implementations of edge detection algorithms are desirable. One of the methods to achieve the speedup in algorithm's execution is to use the potential of available FPGA-circuits. They enable parallel processing of image-data using custom digital structures. There are many methods for edge detection, but most of them can be grouped into two categories, gradient-based and zero-crossing based. The gradient-based methods detect edges by looking for maxima and minima in the first derivative of the image function, usually for the local directional maxima of the gradient magnitude (like in the Canny algorithm [1]). The zero-crossing based methods search for zero crossings in the second derivative of the image in order to localize edges (an example is the Marr-Hildreth algorithm [4]). There are also other, nonlinear methods based on structural properties of the image, like the SUSAN method [7].

Another interesting possibility for edge detection is the morphological gradient, that combines the structural analysis (by using the morphological operators) with some mathematical properties (the image gradient). Morphological gradient operator for a grayscale image  $A$  by using the structural element  $B$  is defined by equation 1, where  $\oplus$  denotes the dilation, and  $\ominus$  denotes the erosion [6]. Further informations on the extension of mathematical morphology operators to the gray-scale images can be found in [3].

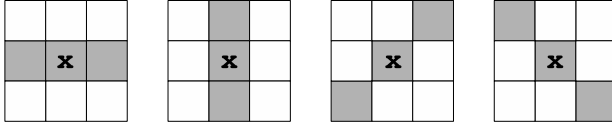
$$MG(A) = (A \oplus B) - (A \ominus B) = \max(A, B) - \min(A, B) \quad (1)$$

The morphological gradient is an attractive tool in some cases because of its properties. The main advantage of the morphological operators is their conceptual simplicity and ease of implementation. The morphological operators rely more on the geometric than mathematical properties of the image. This property helps them to filter out or preserve certain geometric features in the image [2]. Moreover, the morphological gradient does not require normalization to avoid clipping, while the differential convolutional operators often produce the response which is out of the intensity range of the image (clipping). The main drawbacks of the morphological gradient are its sensitivity to noise (rising as the area of the structural element increases) and the lack of edge direction information, which makes the non-maximum suppression impossible. This paper presents methods overcoming these drawbacks while preserving the advantages of morphological gradient edge detection. Additionally, a compact implementation of the algorithm that uses an inexpensive FPGA is described. This implementation is capable of processing the grayscale images of  $512 \times 512$  resolution with the speed reaching hundreds of frames per second.

## 2 Description of the Implemented Algorithm

The regular morphological gradient (see equation 1) provides information about edge strength, but not on the edge orientation. Most edge detectors, like for example the widely known Canny edge detector [1], use the difference filters, such as the Sobel filter, as part of their edge extraction algorithm. Such difference filters are sensitive to edge direction and enable therefore the edge orientation information to be extracted from the filtered image. The knowledge of edge orientation is necessary to perform the non-maximum suppression. Such suppression is necessary, because good quality edges should be placed at the gradient maximum location and also should be thin. To overcome the morphological gradient's insensitivity to the edge direction, it is possible to use direction-biased structuring elements, like those shown in figure 1.

Let us denote the horizontal, vertical and diagonal (45 and 135 degrees) structural elements by  $SE_H$ ,  $SE_V$ ,  $SE_{45}$  and  $SE_{135}$  respectively. The directional gradient for a grayscale image  $A$  is given as the difference between the directional dilation and directional erosion (i.e. dilation and erosion of the image using the



**Fig. 1.** Directional structural elements defined on a  $3 \times 3$  raster. From left to right, respectively:  $SE_H, SE_V, SE_{45}$  and  $SE_{135}$ ; the origin pixel is denoted by  $x$ .

same directional structuring element), and for the four principal directions it is given by equations:

$$G_H(A) = (A \oplus SE_H) - (A \ominus SE_H) \tag{2}$$

$$G_V(A) = (A \oplus SE_V) - (A \ominus SE_V) \tag{3}$$

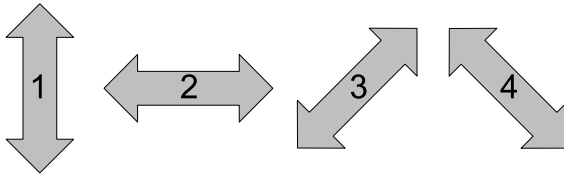
$$G_{45}(A) = (A \oplus SE_{45}) - (A \ominus SE_{45}) \tag{4}$$

$$G_{135}(A) = (A \oplus SE_{135}) - (A \ominus SE_{135}) \tag{5}$$

For any given pixel at the coordinates  $(x, y)$  we define the morphological gradient magnitude as the maximum of the horizontal and vertical gradient filter response at that point:

$$MG_{(x,y)} = \max[G_H(A(x, y)), G_V(A(x, y))] \tag{6}$$

Please note, that combining these two directional gradients corresponds to the computation of a morphological gradient with a cross-like structural element. This is because the morphological operations using complex structural elements can be decomposed into series of operations using simpler structural elements. The knowledge of the magnitudes of directional morphological gradients at any given image point let also the determination of the gradient direction. For performing the non-maximum suppression it is sufficient to classify the edge direction to the one of the four principal directions, see figure 2:



**Fig. 2.** The four principal gradient directions

The edge direction  $\Theta$  can be obtained from the following equation:

$$\Theta_{(x,y)} = \arctan \frac{G_H(A(x, y))}{G_V(A(x, y))} \tag{7}$$

Please note however, that the magnitudes of directional morphological gradients  $G_V(A(x, y))$  and  $G_H(A(x, y))$  are never negative, and therefore the result of the gradient direction computation is not always explicitly the proper angle. However, to classify the local edge direction just to one of the four quadrants, the available information is sufficient. If a diagonal edge is encountered, there are two possible solutions, but comparing both of the diagonal gradient magnitudes allows to discard the improper solution. Therefore, the complete criteria to classify the edge direction of the pixel at coordinates  $(x, y)$  can be formulated as shown in equation 8.

$$dir_{(x,y)} = \begin{cases} 1 & \text{for } \Theta_{(x,y)} \leq 22,5^\circ, \\ 2 & \text{for } \Theta_{(x,y)} > 67,5^\circ, \\ 3 & \text{for } 22,5^\circ < \Theta_{(x,y)} \leq 67,5^\circ \text{ and } G_{45}(A(x, y)) < G_{135}(A(x, y)), \\ 4 & \text{otherwise.} \end{cases} \quad (8)$$

Please note, that the boundary values of  $\Theta_{(x,y)}$  correspond to certain values of  $G_H(A(x, y))$  over  $G_V(A(x, y))$  in the arctangent function (see equation 7). Therefore, the value of the fraction can also be used for edge direction classification. Such approach is favorable, because it eliminates the need to call for the computationally costly arctangent function. Also, it requires less resources than the lookup table approach.

After the gradient magnitude and edge directions are computed, non-maximum suppression can be applied. The non-maximum suppression is done by retaining the points with the gradient magnitude  $MG_{(x,y)}$  higher than some threshold  $t$ , their gradient magnitude being a local maximum in the gradient direction (we assume the gradient direction to be perpendicular to the edge direction). We have observed a significant quality drop on a class of images (mostly artificial test images with strong step edges). The edges were not fully connected, and also some sharp corners were destroyed during the process non-maximum suppression.

The problem can be alleviated by prefiltering the image with a simple  $3 \times 3$  gaussian mask. This approach has also the advantage of filtering some amount of registration noise, and many edge detectors perform such filtering as preprocessing. To that goal we used the pseudo-gaussian mask  $H$  given in equation 9.

$$\mathbf{H} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (9)$$

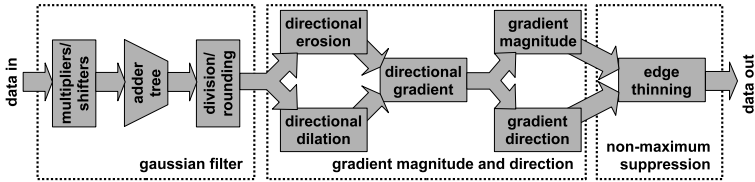
Please note, that all coefficients can be simply implemented by multiplying and dividing by numbers that are the power of 2. Such operations can be easily implemented in hardware by shifting the binary representation of the number left (for multiplication) and right (for division).

### 3 Processor Architecture

Because all the stages of the algorithm operate on a  $3 \times 3$  window, addition of FIFOs with additional address generators working as digital delay lines was



necessary to assure access to all of the nine pixel values in the processed window at once. The FIFOs were implemented using on-chip block RAM. This solution was preferred over the one with external memory for its speed and flexibility. The block RAM was used in the dual-port memory configuration, with an 8-bit interface. This allows to perform a read and write operation in a single clock cycle. The FIFO depth is equal to the horizontal resolution of the image. The whole datapath is organized in pipeline, where all the functional blocks (gaussian filter block, gradient magnitude and direction computation block and the non-maximum suppression block) are connected in series. Thanks to parallel processing, after filling the pipeline, the processor accepts and outputs one byte of data on every clock cycle. The pipeline structure is depicted in figure 3.



**Fig. 3.** Processing pipeline structure (FIFO delay lines have been omitted for clarity)

The use of a gaussian mask given by equation 9 allows to implement multiplication by shifting left the binary representations of numbers. The multiplied values are then added using pipelined adder tree. The result of the addition is then divided by 16 by shifting it right four places. Depending on the values of the discarded bits, the result may be increased by one to achieve proper rounding to the nearest integer value.

The output of the filter is then fed to the block that computes gradient magnitude and direction. In the first clock cycle, all the directional erosions and dilations are computed by selecting the minimum and maximum value under each of the four directional structural elements. In the following clock cycle, the values of directional erosion are subtracted from the respective values of directional dilation. This gives the four directional gradients as the result. Depending on the gradient values, the edge direction is determined in the third and fourth clock cycle. To avoid floating-point computations, the values under arctangent have been split into two integer coefficients. The horizontal and vertical component of the gradient are multiplied by these two coefficients respectively and compared. We have therefore four coefficients, two for each boundary value of  $\theta$ . To reduce the resource cost, one of the coefficients for each boundary value has been set to 128. This allows to implement the multiplication by 128 as bit-shift operation and use two hardware multipliers instead of four. The errors resulting from the differences between the values approximated with the chosen integer values and the actual arctangent values for the boundary values of  $\theta$  do not exceed 0,05%. The values of horizontal and vertical gradient components after multiplication and also the diagonal gradient components are then compared

to classify the edge direction according to equation 8. At this point also, the maximum of horizontal and vertical gradient components is selected as the local gradient magnitude.

The information on edge strength and direction is passed to the non-maximum suppression block. For each pixel with the magnitude above some selected threshold value  $t$ , the magnitude of the pixel is compared to the magnitude of the two pixels along the gradient direction. If the magnitude of this pixel is greater than the magnitudes of the other two pixels, the output is set to '1', otherwise it is set to '0'. The pixels on the output of the non-maximum suppression block form the binary edge image of the input image data.

The whole processor circuit uses six delay lines (two for each functional block), so the delay introduced is equal to six scan lines plus eleven clock cycles (this results from the pipeline depth).

## 4 Implementation and Performance Evaluation

We have implemented the architecture described above in hardware for testing purposes. The processor capable of performing all the operations used in the algorithm was implemented and tested by using VHDL as hardware description language together with Xilinx ISE 8.1 package and Mentor Graphics' ModelSim XE III 6.1e. Target chips were the FPGA's from Xilinx's Spartan 3, namely the XC3S200-4FT256. The design was first tested using simulation software to confirm the correct operation of the post-synthesis and post-place and route model. The design was then implemented in a physical device. We have used the Xplorer script from Xilinx to determine the best implementation settings with the maximum clock speed chosen as the priority. The summary of implementation details is given in table 1.

**Table 1.** Resource usage of the implemented design. The values in percent are given with respect to all resources available.

Parameter	Value
No. of block RAMs	6 (50%)
No. of flipflops	516 (13%)
No. of LUTs	689 (17%)

The software indicates, that the solution is able to operate at pixel clock of 150 MHz. The time to process an 8-bit  $512 \times 512$  image is therefore about 1,8 milliseconds. Functional tests were performed by sending a stream of bytes, representing the image data, to the device and visualizing the data received in response. We have compared the processing time with the performance of the optimized Canny algorithm implemented in the OpenCV library. The tests were run on a PC fitted with 1.6 GHz Turion processor with 1 GB of RAM. The software edge detector turned out to be ten times slower. We have also compared

our design with a FPGA-based edge detection solution that uses the Canny algorithm [5]. The pixel clock of this design is 260 MHz. Our design consumes however significantly less resources and we also believe, that implementing it in a high-end FPGA would result in significant performance increase (up to 2.1 times faster on a Virtex-4 FPGA). Output images are given in figure 4. Threshold level  $t$  was set here to 12. An 8-bit grayscale test image of  $512 \times 512$  resolution has been used. Results of applying the Canny edge detector to the same test image have been provided for comparison. Parameters of the Canny edge detector (threshold levels and standard deviation of the gaussian mask) were as  $t_H = 70$ ,  $t_L = 30$  and mask size  $3 \times 3$ ,  $\sigma = 0.85$ .



(a) Original test image      (b) Morphological edge detector result      (c) Canny edge detector result

**Fig. 4.** Original image and images obtained by applying the morphological and Canny edge detectors

Morphological edge detector demonstrates a decent performance. The localization of edges is good. The edges are also properly thinned. The main problem is the edge streaking. However, this implementation of the edge detector uses only simple thresholding, while the Canny edge detector uses thresholding with hysteresis. Incorporating more sophisticated thresholding schemes is planned in future revisions of the hardware. Another problem is the lack of connectivity at T-junctions and X-junctions. The Canny algorithm however demonstrates similar effects, caused mainly by the gaussian filtering and also the non-maximum suppression.

## 5 Conclusions

A fully functional hardware implementation of the morphological edge detector capable of meeting the demands of real-time applications has been presented in this paper. The use of FPGA circuits offers not only a significant increase in processing speed, but also allows to reduce the cost, size and power consumption of

the device and offers the flexibility to modify existing features and add new features in new hardware revisions. This is an attractive perspective, especially for embedded system. Further research includes the implementation of other feature extraction algorithms, such as the corner detector and image segmentation. We plan to use these feature extraction processors as the part of a system-on-chip processor for high-performance tracking and machine vision applications. Such processor can be incorporated into the flow of visual data-stream, directly after the camera.

## References

1. J Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 8(6):679–698, 1986.
2. Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
3. Henk J. A. M Heijmans. Theoretical aspects of gray-level morphology. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 13(6):568–582, 1991.
4. D. Marr and E. Hildreth. Theory of edge detection. *Proc. Roy. Soc. London*, B207:187–217, 1980.
5. H. Neoh and A. Hazanchuk. Adaptive edge detection for real-time video processing implemented on an FPGA. *Global Signal Processing*, 2004.
6. Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA, 1983.
7. Stephen M. Smith and J. Michael Brady. Susan – a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

---

# Numerical Complexity Reduction in Live-Wire Algorithm

Wojciech Więclawek and Ewa Piętka

Institute of Electronics, Silesian University of Technology, 44-100 Gliwice,  
ul. Akademicka 16  
{wojciech.wieclawek,ewa.pietka}@polsl.pl

**Summary.** In the current study the numerical complexity of the *Live-Wire* (*LW*) method has been significantly reduced. The *LW* technique delineates the edges on the basis of a graph searching algorithm. The size of the graph, traditionally equal to the image size, refers to all image pixels. The graph size reduction can be obtained by withdrawing from the analysis homogeneous regions inside the anatomical structures. In this approach an *FCM* technique has been implemented and tested on selected *MRI* and *CT* examination.

## 1 Introduction

Image segmentation is still an open problem in the field of image processing and computer graphics. Existing algorithms are either sufficiently accurate nor universal in order to be employed to different applications. Complexity of the segmentation problem causes an increase in the number of developed methods. Additionally, proposed solutions often execute many various operations increasing numerical complexity. Simultaneously, users care for receiving fast results. Thus, it seems reasonable to reduce the numerical complexity by withdrawing from the analysis image area of no pertinent information.

This paper is organized as follows: in section 2 the traditional *Live-wire* technique is presented. Then, a modification employing the *FCM* algorithm is introduced (section 3) and the results are shown in section 4.

## 2 Live-Wire Algorithm

*Live-Wire* (*LW*) algorithm [1, 2, 3] is an edge-based segmentation technique and requires manual initialization of characteristic points. Marked points trace the structure boundary at the segmented process. Image boundaries are represented by optimal paths linking the graph vertices, indicated by the user. Hence in the *LW* algorithm image is seen as a weighted, undirected graph  $G = (V, E)$ , where  $V$  denotes the set of vertices and  $E$  denotes the set of edges. Each graph vertex corresponds to a single image pixel and is connected with its eight-neighborhood. Additionally, a suitable weight, assigned to each graph vertex, describes the edge

properties of the image pixel corresponding to this vertex. The full set of weights creates a matrix called an image cost map.

## 2.1 Cost Map Definition

In order to determine, for each image pixel, a value of the membership degree to image boundaries, different methods can be applied. In traditional *LW* method [1, 2, 3] image gradient features are adapted. Among them are: image gradient magnitude, image gradient direction, and laplacian zero-crossing. Other solutions employ the Fourier transform [4, 5] or wavelet image decomposition [6, 7]. The way the cost map is defined influences the segmentation quantity and precision of boundaries delineation. This stage of the cost map analysis has already been discussed in [2, 3, 6, 7].

## 2.2 Graph Searching Algorithm

Detection of the entire structure boundary is carried out in several steps. In each step only a single boundary segment, limited by the pair of user assigned points (the seed and the free point), is found. The distance between the seed point and the free point influencing the delineation accuracy, depends on the boundary curvature. Every part of the boundary structure is connected with the optimal path, which joints vertices representing the seed and the free point.

In order to compute a single boundary segment (any optimal path) a graph searching algorithm is required. In the current study the Dijkstra algorithm [8] has been implemented. It analyzes vertices and their neighborhood step by step, starting from the seed point manually selected by the user. As a result a tree rooted at this point is obtained. This tree constitutes a set of optimal paths, which connects each image pixel with the seed point. In order to select exclusively one path the user has to specified a free point.

In order to compute the next edge segment the graph searching procedure is repeated. Usually, the analysis stops when the entire structure boundary is detected.

## 2.3 LW On-the-Fly

In the traditional *LW* method even the detection of a short boundary part requires the full graph to be searched. One has to keep in mind that the number of vertices is equal to the number of the image pixels and the entire image has to be processed. Additionally, the numerical complexity of the Dijkstra algorithm is proportional to the number of vertices and equals  $\mathcal{O}(n^2)$ , where  $n$  denotes vertices number. Thus, this stage of the *LW* algorithm is the most time consuming. In order to decrease the graph size, the *LW on-the-fly* modification has been proposed [5, 9, 10]. This solution requires a free point specification before the graph searching procedure starts. The searching process stops when the free point is achieved. Therefore, the set of analysed points grows around the seed point in each direction, until the free point is reached.

### 3 LW-FCM Algorithm

Both solutions: the traditional *LW* method and the *LW on-the-fly* take into consideration each image pixel independent of its properties and context. From the image edge-based segmentation point of view pixels located in homogeneous areas can be ignored, because they have no boundary features required in the *LW* methodology. Thus, the *FCM* algorithm is employed in order to find image pixels to be subjected to the graph searched phase. The image pixels described in terms of the gray level value, is classified into  $c$  classes. In the current study the class number has been determined empirically. The maximum of corresponding elements of the partition matrices allows a pixel to be assigned to a certain class. As a result, each image pixel is referred to a class label (value from  $1 \dots c$ ). In this image, pixels located in homogeneous areas have got the same class label and can be skipped during the graph searching process. Only pixels, whose neighborhood belongs to a different class are located at or near an edge and only these pixels are subjected to the graph search process.

Additionally, the number of classes influences the sensitivity of the edge detection. For small  $c$  only strong edges are detected, whereas large  $c$  enables a correct segmentation of structures surrounded even by weak and thin boundaries. It also influences the numerical complexity of *LW-FCM* method (as shown in section 4).

#### 3.1 LW-FCM On-the-Fly

The *LW-FCM* algorithm can be performed in the *on-the-fly* mode. Thus, the processing breaks when the initially indicated by the user free point is reached. The benefits of this modification strongly depends on the distance between the seed and the free point.

## 4 Results

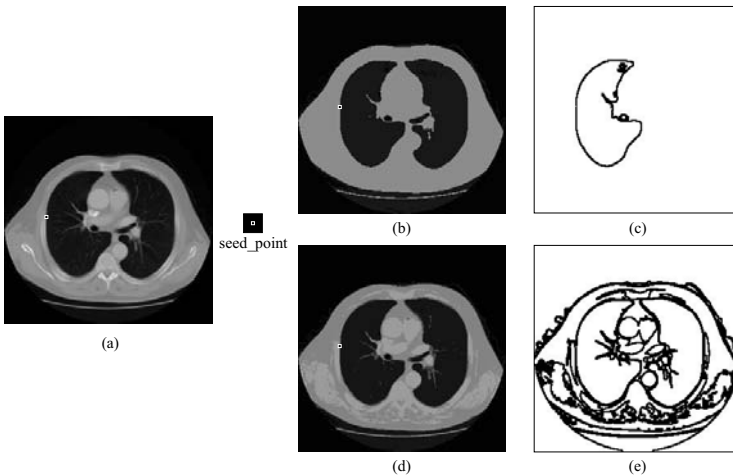
To access the accuracy of the segmentation procedures as well as the numerical complexity of the algorithms, the following evolution studies have been conducted. The computed tomography (*CT*) chest and magnetic resonance imaging (*MRI*) knee studies have been selected. The lung areas in the first and the bony structures in the second type of exams have been segmented.

First the numerical complexity of the traditional *LW* and the modified *LW-FCM* algorithms have been tested. It is strongly influenced by the graph searching algorithm (Dijkstra algorithm [8]), whose numerical complexity depends on the image pixel number building the graph. With this regard estimation of the numerical complexity value will be express in pixels number or the percentage of their total number in the image (Table 1). In the traditional *LW* algorithm the entire image (100%) is subjected to the graph search phase. On the contrary the *LW-FCM* method depends on number of classes  $c$  in the *FCM* algorithm. Additionally, the number of analyzed pixels in *LW-FCM* algorithm strongly depends on the segmented structure and the seed points marked by the user.

**Table 1.** Numerical complexity of the *LW* algorithm in comparison with *LW-FCM* for various number of classes  $c$

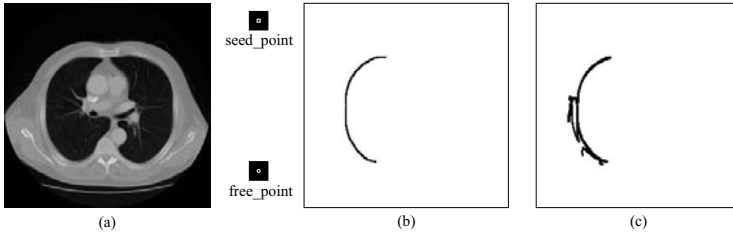
Method	Number of pixels	%
Traditional	65536	100
<i>LW</i> method		
<i>LW-FCM</i> method, $c = 2$	1116	1, 7
<i>LW-FCM</i> method, $c = 3$	1265	1, 93
<i>LW-FCM</i> method, $c = 4$	12324	18, 8
<i>LW-FCM</i> method, $c = 5$	15088	23, 02
<i>LW-FCM</i> method, $c = 6$	16157	24, 65
<i>LW-FCM</i> method, $c = 7$	17889	27, 29

In Table 1 a rapid increase in the number of pixels building the graph can be noticed. Another interesting remark can be made by comparing a small increase in the edge points when  $c$  changes between 5 and 7. However, only less than 30% of the image points are required to sufficiently extract even small edges. The clustering results and pixels building the graph are shown in Fig. 1.



**Fig. 1.** *LW-FCM* method, (a) original image with seed point, (b), (d) clustered version of image for  $c = 3$  and  $c = 5$ , respectively, (c), (e) image pixels involved in the graph search procedure





**Fig. 2.** *LW-FCM-on-the-fly* method, (a) original image with seed and free point, (b),(c) image pixels involved in the graph search procedure for  $c = 3$  and  $c = 5$ , respectively

Similar analysis has been conducted for the *LW on-the-fly* and *LW-FCM on-the-fly* algorithms. Since these techniques strongly depend on the seed point marked by the user, the increase in the number of classes does not effect the numerical complexity. Fig.2 shows the image pixels examined at the graph searched phase for  $c = 3$  and  $c = 5$ . Results (Table 2) indicate, the *on-the-fly* modification of the *LW* and *LW-FCM* methods to be more effective than the traditional *LW* and *LW-FCM* approach, respectively. In all structures the characteristic points

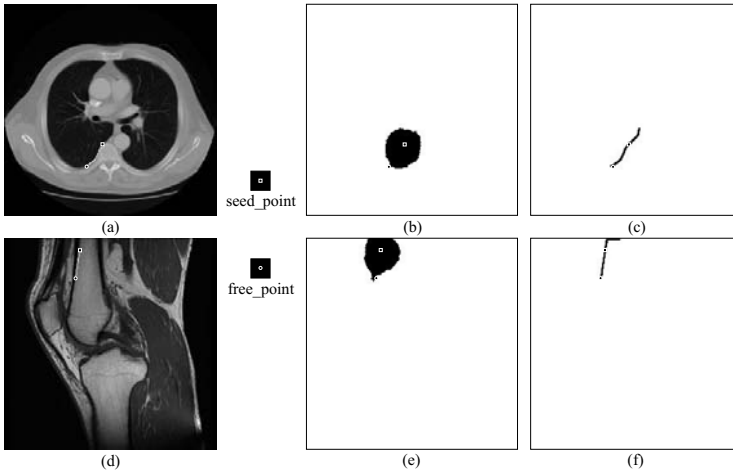
**Table 2.** Numerical complexity of *LW on-the-fly* algorithm in comparison with *LW-FCM on-the-fly* for various number of classes  $c$

Method	Number of pixels	%
<i>LW on-the-fly</i> method	12786	19, 51
<i>LW-FCM on-the-fly</i> method, $c = 2$	447	0, 68
<i>LW-FCM on-the-fly</i> method, $c = 3$	685	1, 04
<i>LW-FCM on-the-fly</i> method, $c = 4$	710	1, 08
<i>LW-FCM on-the-fly</i> method, $c = 5$	875	1, 33
<i>LW-FCM on-the-fly</i> method, $c = 6$	1084	1, 65
<i>LW-FCM on-the-fly</i> method, $c = 7$	1033	1, 57

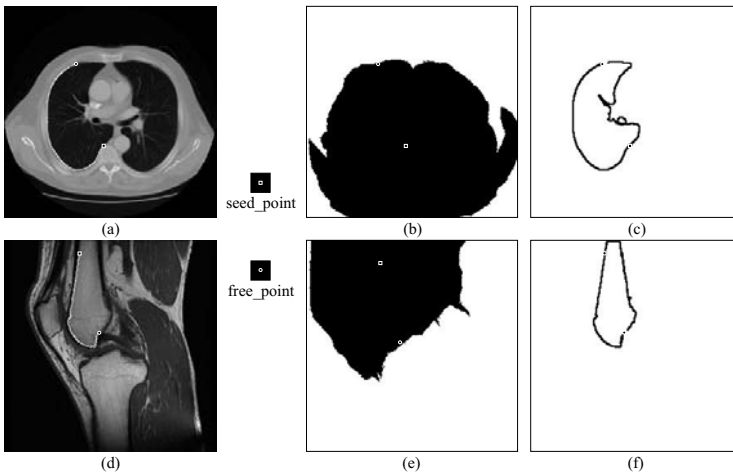
**Table 3.** Numerical complexity of *LW on-the-fly* algorithm comparison with *LW-FCM on-the-fly* for distance between characteristic points

Method	CT chest examination		MRI knee joint examination	
	Number of pixels	%	Number of pixels	%
<i>LW on-the-fly</i> method	1659	2, 53	1840	2, 81
<i>LW-FCM on-the-fly</i> method	146	0, 22	148	0, 22
<i>LW on-the-fly</i> method	37387	57, 05	21445	32, 72
<i>LW-FCM on-the-fly</i> method	1107	1, 69	750	1, 14

have been located identically. Since in the *on-the-fly* mode, results depend on the geometrical distance between the seed point and the free point the *LW on-the-fly* and *LW-FCM on-the-fly* have also been compared (Table 3). Graphical representation is presented in Fig. 3, Fig. 4.



**Fig. 3.** Numerical complexity reduction of *LW* algorithm for short boundary segments, (a), (d) original images with seed and free point, (b), (e), (c), (f) image pixels involved in the graph search procedure for: (c), (d), *LW-on-the-fly* and (e), (f) *LW-FCM-on-the-fly*



**Fig. 4.** Numerical complexity reduction of *LW* algorithm for long boundary segments, (a), (d) original images with seed and free point, (b), (e), (c), (f) image pixels involved in the graph search procedure for: (c), (d), *LW-on-the-fly* and (e), (f) *LW-FCM-on-the-fly*

The evaluation of the methodology has also been performed on a set of 50 chest *CT* and 25 knee joint *MRI* images. The *LW-FCM* approach reduces the number of analyzed pixels to down to 2 – 31% depending on the number of

**Table 4.** Image pixels involved in the graph searching procedure in *LW* and *LW-FCM* methods

Image	Structure	Number of pixels %	
		<i>LW</i>	<i>LW-FCM</i>
		$c = c_{opt}$	$c = 5$
CT chest examination	single lung	100	1,7029 23,022
MRI knee joint	thighbone	100	1,1765 31,801
MRI knee joint	tibia	100	5,5038 31,801

**Table 5.** Image pixels involved in the graph searching procedure in *LW on-the-fly* and *LW-FCM on-the-fly*

Image	Structure	Number of pixels %	
		<i>LW</i>	<i>LW-FCM</i>
		$c = c_{opt}$	$c = 5$
CT chest	single lung	20,108	0,58441 0,85754
MRI knee joint	thighbone	11,214	0,4837 2,4765
MRI knee joint	tibia	31,313	3,5873 15,039

classes (Table 4). Another decrease of the analyzed area is obtained when the *on-the-fly* mode is employed (Table 5).

Apart from the reduction of the numerical complexity an improve in the segmentation accuracy has been obtained. Since the *FCM* phase excludes the homogeneous regions from the graph search step only the edge of an anatomical structure can be followed. This means that no short cuts are possible. This decreases the required number of characteristic points and reduces the user interaction Table 6.

**Table 6.** Number of characteristic points

Image	Structure	Number of points	
		<i>LW</i>	<i>LW-FCM</i>
CT chest examination	single lung	6 – 8	2 – 3
MRI knee joint	thighbone	7 – 16	3 – 5
MRI knee joint	tibia	5 – 8	2 – 5

## 5 Conclusion

The introduced *LW-FCM* method reduced the numerical complexity of the traditional *LW* technique. User interaction can be decreased to only a few points

to be marked on the edge. Finally, an improvement of the accuracy has been obtained. The method has been evaluated as a image segmentation tool for selected soft tissue and bonny structures.

## References

1. Mortensen E N, Barrett W A (1995) Intelligent Scissors for Image Composition. *Computer Graphics and Interactive Techniques (ACM SIGGRAPH '95)* 191–198
2. Barrett W A, Mortensen E N (1997) Interactive Live-Wire Boundary Extraction. *Medical Image Analysis* 1:4:331–341
3. Mortensen E N, Barrett W A (1998) Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing* 60:5:349–384
4. O'Donnell L, Westin C F, Grimson W E L, Ruiz-Alzola J, Shenton M E, Kikinis R (2001) Phase-Based User-Steered Image Segmentation. *Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention* 1022–1030
5. O'Donnell L (2001) *Semi-Automatic Medical Image Segmentation*. MA Thesis, Massachusetts Institute of Technology, Massachusetts
6. Więcasawek W (2006) Live-Wire Method with Wavelet Cost Map Definition for MRI Images. *IFAC Workshop on Programmable Devices and Embedded Systems (PDeS' 2006)* 197–202
7. Więcasawek W (2006) Live-Wire Method With FCM Classification. *Mixed Design of Integrated Circuits and Systems (MIXDES'2006)* 756–760
8. Dijkstra E W (1959) A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* 1:269–270
9. Falcão A X, Udupa J K, Miyazawa F K (2000) An Ultra-Fast User-Steered Image Segmentation Paradigm: Live Wire on the Fly. *IEEE Transactions on Medical Imaging* 19:1:55–62
10. Kang H W, Shin S Y (2002) Enhanced Lane: Interactive Image Segmentation by Incremental Path Map Construction. *Graphical Models* 64:5:282–303

---

# Image Segmentation Using Adaptive Potential Active Contours

Arkadiusz Tomczyk

Institute of Computer Science, Technical University of Lodz, Wolczanska 215,  
90-924, Lodz, Poland  
tomczyk@ics.p.lodz.pl

**Summary.** In the paper a new segmentation method is proposed. *Adaptive potential active contour approach* (APAC) is a result of the relationship between active contour methods and classifier construction techniques. As an optimization algorithm simulated annealing was used to avoid local minima of energy function. The presented approach allows to obtain contours of various topology and a new adaptation mechanism can be introduced to improve segmentation results.

## 1 Introduction

Image segmentation is a crucial element of every system (e.g. medical diagnostic systems) where image recognition is necessary. There exist many different techniques of image segmentation such as thresholding methods, region growing and splitting ([8]), etc. All of them, however, possess limitations in utilization of high-level knowledge which can be obtained from experts and which can significantly improve results of object identification. As a solution of that problem active contour techniques were proposed ([1, 2, 3, 4]). In this paper a new method from that group is presented.

The paper is organized as follows: in section 2 relationship between active contour methods and classifier construction is presented, in section 3 a new *potential active contour* (PAC) method is described, section 4 is devoted to adaptation mechanism and finally the last two sections contain examples and summary of the proposed method, respectively.

## 2 Active Contours and Classifier Construction

### 2.1 Classifiers

Classifiers are pattern recognition methods where the set of recognition results of objects  $\mathcal{O}$  is finite. Those results from set  $\mathbb{L}(\mathcal{L})$  are usually called labels (further  $\mathbb{L}(\mathbb{N})$  will denote a set  $\{1, \dots, \mathbb{N}\}$  for  $\mathbb{N} \in \mathbb{N}$  and  $\mathcal{L}$  will denote the number of possible labels). Of course, none of the classification algorithms can operate on the objects directly. They are able to operate on data extracted from object (i.e. sensed, measured etc.). Consequently, there must exist some method of feature

extraction. Summing up those considerations, classifiers can be considered as functions  $k : \mathcal{X} \rightarrow \mathbb{L}(\mathbb{L})$  that assign proper labels to the objects described by means of their features from space  $\mathcal{X}$ .

The actual problem is construction of classification functions. There are many classifiers  $k \in \mathcal{K}$  that map  $\mathcal{X}$  into  $\mathbb{L}(\mathbb{L})$  (where  $\mathcal{K}$  is a set of all possible *classifiers* in a given problem). To enable a choice of the proper one additional information (expert knowledge) must be used (e.g. a *training set* with correctly labeled objects). This knowledge can be formulated in the form of performance index  $Q : \mathcal{K} \rightarrow \mathbb{R}$  capable to evaluate the usefulness of each classifier. In consequence, the problem of classification can be expressed as optimization of certain performance index.

There are many practical methods of classifier construction. Some of them bases on the quite obvious assumption, coming from observation of human recognition process, that the result of classification depends on the labels of known objects similar to the currently recognized one. Here the *k-NN* and *potential function* methods can be given as an example ([5, 7]). The other approaches, e.g. *neural networks*, try to simulate the human brain activity ([5, 6, 7]), etc.

## 2.2 Active Contours

Active contour methods were firstly introduced in [1]. The main idea was to find an optimal contour  $c \in \mathcal{C}$  describing object in the image (where  $\mathcal{C}$  denotes space of all considered *contours*). Optimality criterion was the so called energy function  $E : \mathcal{C} \rightarrow \mathbb{R}$  evaluating contours (internal energy was able to evaluate contour's shape while external - its location in the given image). The difference of this approach from other segmentation methods is the utilization of high-level knowledge for detection of object composed of lower-level image elements (pixels).

In [1] contour is defined as a parametric curve and consequently the energy function is in fact a functional. To minimize it calculus of variations is used which transforms it into the problem of iterative solution of partial differential equations set (discretization of contour was necessary). However, since the first description of the *snakes* method there appeared a variety of other active contour techniques. They differ in the method of contour description, in formulation of the energy function and finally in the way of contour evolution. Examples are: *active shape models* where the contour (polygon) can only change within the previously trained model of shape ([2]), *geodesic active contours* where contour is defined as an intersection of 3D surface and some plane which allows to consider contours of different topology ([3]) and *Brownian strings* where contour is described by means of chain code, external energy is statistically trained and for evolution simulated annealing is used ([4]). This short review presents only a small number of methods but it shows their diversity.

## 2.3 Relationship

In [10] relationship between active contour methods and construction of classifiers was described. It was shown that contours can be interpreted as contextual

classifiers of pixels where context of currently classified pixel are other pixels from its neighborhood. Thus, the recognition result depends on the features of all those pixels allowing to consider relations between them. Consequently also, the search of an optimal contour can be interpreted as a method of optimal classifier construction where energy  $E$  is performance index  $Q$  (in both cases during optimization the parameters of the assumed contour or classifier model, respectively, are sought).

The described relationship allows to exchange the methods and algorithms ([11, 12]) between active contours and classifier construction methods described in sections 2.1 and 2.2 respectively. Example are *adaptive potential active contours* (APAC). This technique is a new intuitive method of image segmentation which possesses the advantages of active contour methods but originates from *potential function* method so far used only as a pattern recognition method.

### 3 Potential Active Contours

#### 3.1 Potential Function Method

*Potential function* method ([5]) is one of those methods of pattern recognition, which assumes that label of the classified object should depend on the labels of currently known, similar objects. The similarity measure considered in this case is distance function  $\rho : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  and consequently the method can be applied in any metric space  $\mathcal{X}$ .

Let  $\mathcal{D}^l = \{\mathbf{x}_1^l, \dots, \mathbf{x}_{N^l}^l\}$  denote a subset of  $\mathcal{X}$  containing  $N^l$  control objects corresponding to the label  $l \in \mathbb{L}(\mathbb{L})$  and let the function  $P : \mathbb{R} \rightarrow \mathbb{R}$  denote the potential function. Example are exponential potential function:

$$P_{\Psi, \mu}^{exp}(d) = \Psi e^{-\mu d^2} \quad (1)$$

and inverse potential function:

$$P_{\Psi, \mu}^{inv}(d) = \frac{\Psi}{1 + \mu d^2} \quad (2)$$

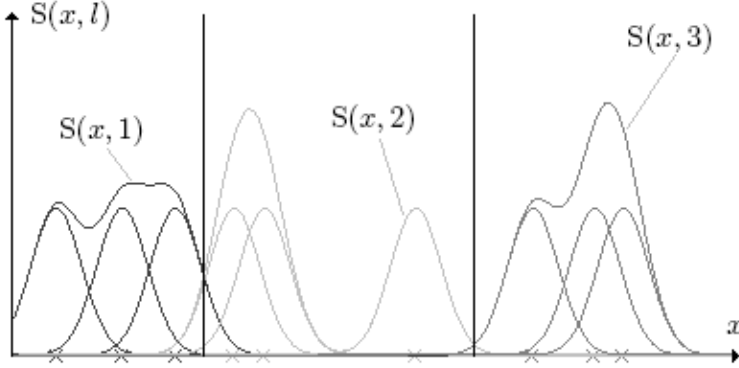
where  $\Psi$  and  $\mu$  are parameters controlling the maximum strength of the potential field in its center and its distribution, respectively (Fig. 1). The *potential function* classifier  $k^{pot} : \mathcal{X} \rightarrow \mathbb{L}(\mathbb{L})$  is defined in the following way:

$$k^{pot}(\mathbf{x}) = \arg \max_{l \in \mathbb{L}(\mathbb{L})} S(\mathbf{x}, l) \quad (3)$$

where  $\mathbf{x} \in \mathcal{X}$ ,  $S : \mathcal{X} \times \mathbb{L}(\mathbb{L}) \rightarrow \mathbb{R}$  and  $S(\mathbf{x}, l)$  denote the resulting potential of object  $\mathbf{x} \in \mathcal{X}$  in the field generated by objects from set  $\mathcal{D}^l$ :

$$S(\mathbf{x}, l) = \sum_{i=1}^{N^l} P_{\psi_i^l, \mu_i^l}(\rho(\mathbf{x}_i^l, \mathbf{x})) \quad (4)$$

To put it differently, the object  $\mathbf{x} \in \mathcal{X}$  receives label  $l \in \mathbb{L}(\mathbb{L})$  if the summary potential of objects from  $\mathcal{D}^l$  has the maximum value (Fig. 1).



**Fig. 1.** The  $l$  resulting potential fields generated in space  $\mathcal{X} = \mathbb{R}$  by control points  $\mathcal{D}^l$  and the boundary dividing that space into  $\mathcal{X}^l$  regions where  $l \in \mathbb{L}(\mathbb{L})$  and  $\mathbb{L} = 3$  (the inverse potential function and Euclidean metric were used)

### 3.2 Potential Contour

Considering any classifier  $k$  in metric space  $\mathcal{X}$  it can be observed that it divides that space into  $\mathbb{L}$  regions  $\mathcal{X}^l = \{\mathbf{x} \in \mathcal{X} : k(\mathbf{x}) = l\} \subseteq \mathcal{X}$  for  $l \in \mathbb{L}(\mathbb{L})$  containing objects with different labels and that there are such objects in  $\mathcal{X}$  that separates those regions (Fig. 1).

While segmenting the image only two classes ( $\mathbb{L} = 2$ ) are considered (label 1 will be treated as background label and 2 as object label) and pixels are the objects that are to be classified. Let  $I : \mathbb{L}(\mathbb{W}) \times \mathbb{L}(\mathbb{H}) \rightarrow \mathbb{L}(\mathbb{I})$  denote a gray-scale image of width  $\mathbb{W}$  and height  $\mathbb{H}$  with  $\mathbb{I}$  intensity levels (color images can also be considered) and let  $\mathcal{X} = \mathbb{R}^2$  (of course  $\mathbb{L}(\mathbb{W}) \times \mathbb{L}(\mathbb{H}) \subset \mathcal{X}$ ). Any classifier  $k : \mathcal{X} \rightarrow \mathbb{L}(\mathbb{L})$  defined in space  $\mathcal{X}$  divides it into two separate regions  $\mathcal{X}^l$  for  $l \in \mathbb{L}(\mathbb{L})$ . Those regions can be used to define background  $\mathcal{X}^1 \cap \mathbb{L}(\mathbb{W}) \times \mathbb{L}(\mathbb{H})$ , object  $\mathcal{X}^2 \cap \mathbb{L}(\mathbb{W}) \times \mathbb{L}(\mathbb{H})$ , and the boundary describing contour  $c$  in the image:

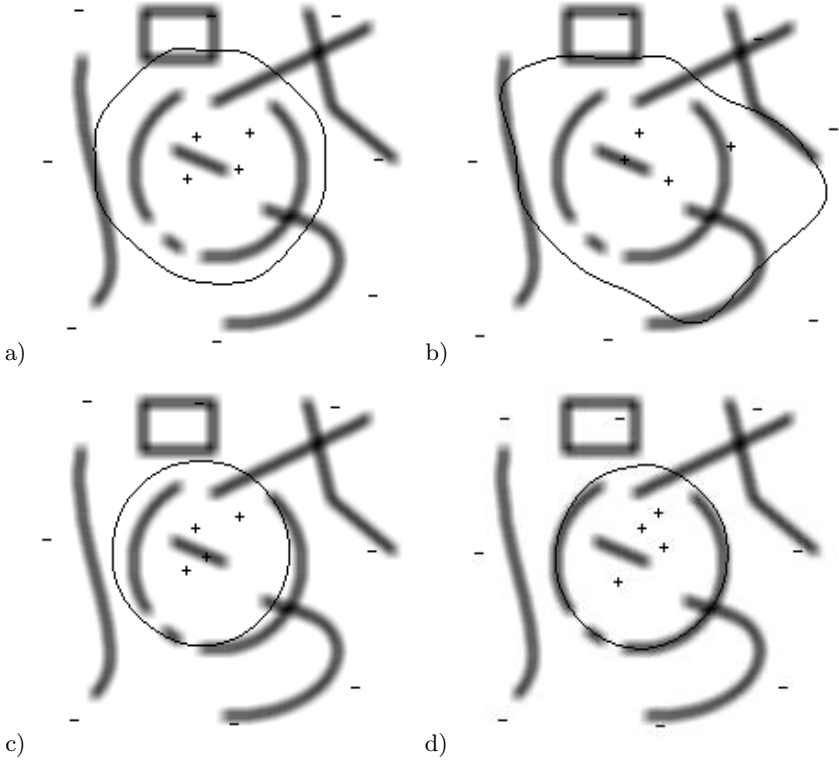
$$\mathcal{B} = \{\mathbf{x} \in \mathbb{L}(\mathbb{W}) \times \mathbb{L}(\mathbb{H}) : \mathbf{x} \in \mathcal{X}^2 \wedge N(\mathbf{x}) \cap \mathcal{X}^1 \neq \emptyset\} \tag{5}$$

where  $N(\mathbf{x}) = \{\langle i-1, j \rangle, \langle i+1, j \rangle, \langle i, j-1 \rangle, \langle i, j+1 \rangle\} \subset \mathbb{L}(\mathbb{W}) \times \mathbb{L}(\mathbb{H})$  for  $\mathbf{x} = \langle i, j \rangle \in \mathbb{L}(\mathbb{W}) \times \mathbb{L}(\mathbb{H})$  (because of practical reasons it is assumed that for coordinates that lie outside the image the classifier returns 1 and thus  $\mathcal{B} \subset \mathcal{X}^2 \cap \mathbb{L}(\mathbb{W}) \times \mathbb{L}(\mathbb{H})$ ). Contour obtained for  $k^{pot}$  classifier is called a *potential contour*. Its shape for a given metric  $\rho$  and potential function  $P$  depends on the distribution of control points in sets  $\mathcal{D}^1$  and  $\mathcal{D}^2$  and on the parameters of potential functions (Fig. 2, Fig. 3).

### 3.3 Energy of Potential Contour

To present the concept of *active potential contours* a few simple energy functions can be proposed:





**Fig. 2.** Sample results of segmentation using *potential active contour* method ( $N^1 = 8$ ,  $N^2 = 4$ ): (a) - initial contour, (b) - result of evolution where only  $E^{int}$  was used, (c) - result of evolution where only  $E^{circ}$  was used, (d) - result of optimization where both energies were combined (their weighted sum was used)

- intensity energy which tries to place contour in darker areas of the image:

$$E^{int}(c) = \sum_{\langle i,j \rangle \in \mathcal{B}} I(i,j) \quad (6)$$

- homogeneity energy which tries to find contour that describes object composed of pixels with given intensity  $T \in \mathbb{L}(I)$ :

$$E^{hom}(c) = \sum_{\langle i,j \rangle \in \mathcal{X}^1} \delta(I(i,j), T) + \sum_{\langle i,j \rangle \in \mathcal{X}^2} (1 - \delta(I(i,j), T)) \quad (7)$$

- circularity energy  $E^{circ}$  that tries to evaluate the difference between contour and the the closest circle which can be found using coordinates from the set  $\mathcal{B}$ .

The first two energies are external energies (they evaluate the position of contour in the image) while the last one is an internal energy (it evaluates contour shape).

Of course, they can be applied together - the resulting energy is a weighted sum of them.

### 3.4 Potential Contour Evolution

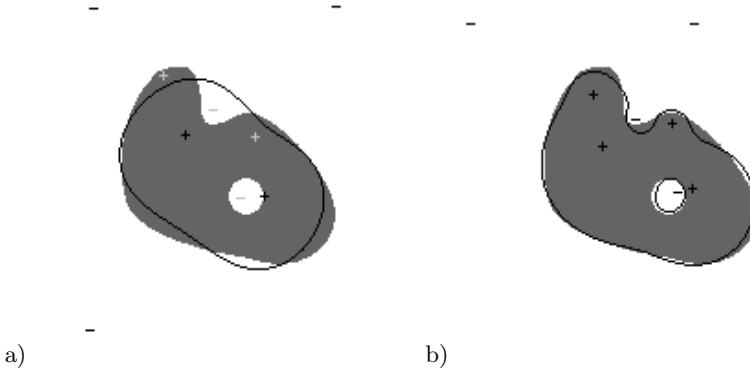
To find the optimal shape of *potential contour*  $c$  the optimal parameter vector of length  $4(N^1 + N^2)$  must be found. Different optimization techniques can be used. So far two iterative methods were tested: *steepest descent* method and *simulated annealing* method ([9]). In this paper, only the second one with exponential cooling scheme and the solution generator randomly changing the components of parameter vector is used (the first one needs the gradient of energy function that must be estimated by means of finite differences which can be time-consuming). Other optimization techniques, e.g. genetic algorithm, can be also of use here.

## 4 Adaptation

Not only the values of parameters describing *potential contour* are important. Also the proper choice of the number of control points describing background  $N^1$  and object  $N^2$  is crucial because it determines the class of shapes that can be obtained during the optimization. If that number is too small some of the shapes can be unreachable and if the number is too large it can significantly slow the optimization process as well as it can cause that the global optimum will be hard to reach. That is why the adaptation mechanism of *potential active contour* is also proposed in this paper (APAC). This mechanism allows to begin the contour evolution with a small number of control points and to find an approximate position of the contour. Later, new control points can be added there where the shape is not satisfactory. It allows to increase the flexibility of the contour in those regions and consequently it allows to obtain better segmentation results.

## 5 Examples

To illustrate the described in previous sections concepts two examples can be considered. In Fig. 2 the first one is presented where intensity ( $E^{int}$ ) and circularity ( $E^{circ}$ ) energies were used. It shows the influence of choice of the energy on segmentation results. If only the external energy is used (Fig. 2b) the contour finds only the best position in the image which in the presence of many other objects (or noise) is usually not satisfactory. Similarly if only internal energy is applied contour neglects all the information from the image and evolves independently (Fig. 2c). This example shows also that *active potential contours* like other active contour methods allow to localize object even if the information in the image is incomplete (the circle is not continuous) and even if there are other objects that can potentially hinder the segmentation. The second example (Fig. 3) shows the influence of adaptation mechanism on segmentation results (only ( $E^{hom}$ ) was used). New control points are added there where it allows to locally improve contour shape and consequently its energy. This example shows



**Fig. 3.** Adaptation mechanism (only  $E^{hom}$  with  $T = 100$  was used): (a) - first result of contour evolution ( $N^1 = 4$ ,  $N^2 = 2$ ) and added control points, (b) - result after adaptation ( $N^1 = 6$ ,  $N^2 = 4$ )

also that *adaptive potential active contours* can be used to find objects of different topology.

## 6 Summary

In the paper a new segmentation method, namely *adaptive potential active contours*, is proposed. It originates from the relationship between active contour techniques and methods of optimal classifier construction. This relationship allows to use the classic *potential function* method for contour description. The presented approach has the same advantages as other active contour techniques (it allows to incorporate any expert knowledge into the energy function and consequently it allows to avoid the necessity of explicit feature extraction). Additionally, it is intuitive because it bases on the same analogies as the *potential function* method. Moreover, an additional adaptation mechanism can be used to improve segmentation results (it dynamically changes the class of shapes that can be considered during optimization). The presented method can be also easily generalized for other classification tasks where more complex objects than pixels are considered. Those aspects have already been described in [12, 13] and will be further investigated together with *adaptive potential contour method*. In particular, the ability of usage of any expert knowledge for classifier construction seems to be very promising.

## References

1. Kass M., Witkin W., Terzopoulos D.: Snakes: Active Contour Models. *International Journal of Computer Vision*. (1988) 321–331.
2. Cootes T., Taylor C., Cooper D., Graham J., Active Shape Model - Their Training and Application, *CVGIP Image Understanding*, (1994) 61(1) 38–59.

3. Caselles V., Kimmel R., Sapiro G.: Geodesic Active Contours. *International Journal of Computer Vision* 22(1) (1997) 61–79.
4. Grzeszczuk R., Levin D., Brownian Strings: Segmenting Images with Stochastically Deformable Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1997) vol. 19 no. 10 1100-1013.
5. Tadeusiewicz R., Flasiński M., *Pattern Recognition*, PWN, Warsaw, (1991) (in Polish).
6. Bishop Ch.: *Neural Networks for Pattern Recognition*. Clarendon Press. Oxford (1993).
7. Looney C.: *Pattern Recognition Using Neural Networks. Theory and Algorithms for Engineers and Scientists*. Oxford University Press. New York (1997).
8. Gonzalez R., Woods R., *Digital Image Processing*, Prentice-Hall Inc., New Jersey (2002).
9. Kirkpatrick S., Gelatt C. D. Jr., Vecchi M.P.: Optimization by Simulated Annealing. *Science* 220. (1983) 671-80.
10. Tomczyk A., Szczepaniak P. S.: On the Relationship between Active Contours and Contextual Classification. *Computer Recognition Systems. Proceedings of the 4th International Conference on Computer Recognition Systems, CORES'05*. Poland. Springer Berlin, Heidelberg, New York. (2005) 303–311.
11. Tomczyk A.: Active Hypercontours and Contextual Classification. *Intelligent Systems Design and Applications Proceedings of 5th International Conference on Intelligent Systems Design and Applications, ISDA'05*. IEEE Computer Society Press. (2005) 256–261.
12. Tomczyk A., Szczepaniak P. S.: Adaptive Potential Active Hypercontours. *8th International Conference on Artificial Intelligence and Soft Computing (ICAISC)*. Springer-Verlag Berlin, Heidelberg. (2006) 692–701.
13. Tomczyk A., Szczepaniak P. S.: Contribution of Active Contour Approach to Image Understanding. *Proceedings of the 2007 IEEE International Workshop on Imaging Systems and Techniques (IST)*. Krakow. Poland, ISBN: 1-4244-0965-9 (2007).

---

# Towards an Exploration of GCA Ordered Pixels

Maria Grzegorek

Institute of Computer Science, Polish Academy of Sciences, Ordona 21, 01-237  
Warsaw  
mary@ipipan.waw.pl

**Summary.** An interesting property of GCA (Gradient Correspondence Analysis) re-ordered pixels has been revealed. Pixels of gray level image are described by their gray level, by calculated gradient module and a family of calculated variables taking into account behavior of pixels' modules in the neighborhood of the pixel. Evolved dataset is processed by GCA procedure and forms new ordering of pixels. Line chart for modules shows *quills* - lines consisting of pixels with the same values of the variables family. Pixels in the line chart for gray level lay along distinct curves and can be extracted with the aim of gradient module.

**Keywords:** Image, gray level, gradient module, Grade Correspondence Analysis.

## 1 Introduction

Images can be processed using grade methods like every phenomenon. Grade models and methods were extensively described in [1]. Program GradeStat, which has been developed in ICS PAS, implements these methods and can be downloaded as free software [2]. Many examples of data grade exploration in medical and demographic domains are described in [3] and [4]. Multivariate data originated from images and processed involving GCA were explored in [5] and [6].

In Section 2 there is shortly explained dataset forming. Section 3 mentions how reordering of rows and columns of the dataset is made. Section 4 presents the revealed properties of line charts of the pixels rearranged according to GCA. Line charts including quills are considered in [7] from a point of view of data analysis. Closing Section 5 suggests some changes concerning forming of the variables.

## 2 From an Image to a Dataset

The gray level image is a source of a set of variables. Pixel in the image has its gray level value  $gm$ , usually within range  $0 \div 255$ . The second variable is gradient module  $gm$  which reflects changes in the gray level in a neighborhood of the pixel. Sobel operator is chosen [8]. The variable  $gm$  preserves to some extent space connections between neighboring pixels. More space dependencies renders a family of variables. This group of variables is constructed using base

parameter  $b$  defined as a fixed fraction of a maximal value of the gradient module  $gm_{max}$ . The maximal module varies from image to image and therefore product  $b \times gm_{max}$  seems to be suitable to establish the variables family. Thresholds  $t_i, i = 1, \dots, k$  are calculated for consecutive variables  $n_i, i = 1, \dots, k$  belonging to the variables family. Thresholds are equal to fractions of  $gm_{max}$ :  $t_i = i \times b \times gm_{max}$ . They are applied to compare gradient modules of neighboring pixels.

Each pixel in the image is surrounded by its neighbors. The distance between the values of the pixel's gradient module and the gradient module of neighboring pixel is compared with the threshold and it is checked how many neighboring pixels have their distance less than the value of this threshold. The value of the variable is equal to the number of neighboring pixels fulfilling thresholding criterion.

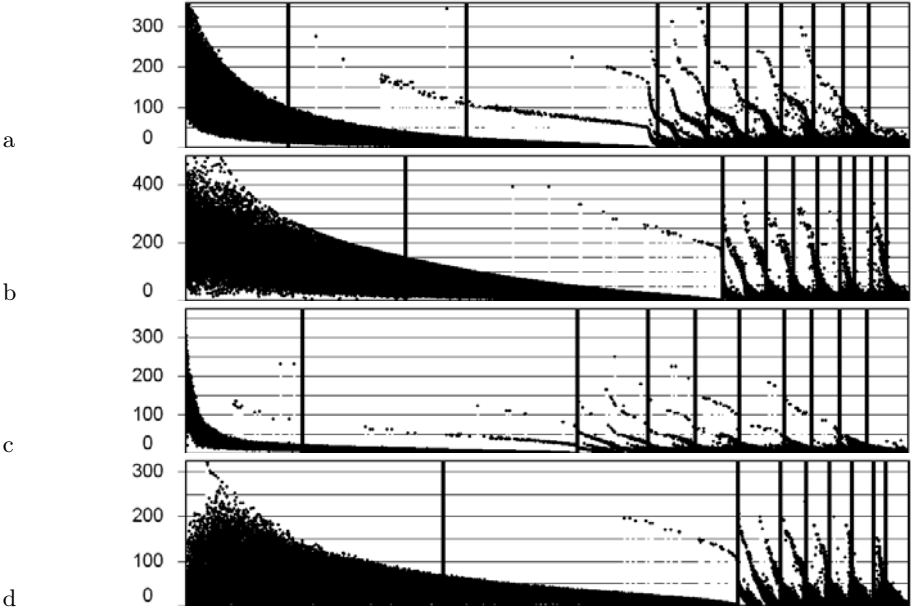
Sequence of variables is constructed with the aim of consecutive thresholds, with chosen value  $b$  and fixed size of neighborhood.

### 3 GCA Reordering of Pixels Dataset

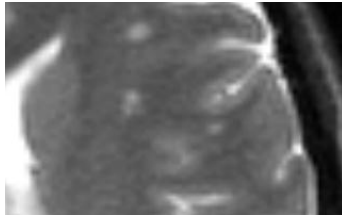
The dataset resulting from pixels is then processed by the GCA procedure. The multivariate data matrix consists of rows corresponding to pixels and columns corresponding to variables which describe different features of pixels. The matrix is normalized to obtain an overrepresentation matrix: each cell is divided by the total of the column to which the cell belongs and it is divided by the total of a row. A chart called overrepresentation map is a very useful in exploration of multivariate datasets [1][2]. The GCA procedure reorders rows and columns of normalized matrix to obtain possibly maximal rows-columns positive dependence measure (for instance Spearman's rho). This ensures that adjacent rows (pixels) are rather similar one to another. The GCA procedure is supplemented with post GCA clustering which groups adjacent pixels in a number of disjoint subsets. This subsets can be visualized in a separate images and they constitute decomposition of the image onto subimages [4][5].

Line charts (called stripcharts in GradeStat) are another way to present GCA reordered dataset. Figure 1 shows the line charts of the variable ( $gm$ ) for four different images. On a horizontal axis are pixels ordered according to GCA. Vertical lines separate clusters. Ten clusters are chosen in each example. There are different numbers of pixels distributed to clusters. On a vertical axis is a value of the gradient module. Maximal values of  $gm$  in the images are different. The parameter  $b$  is equal to 0.2. The first  $gm$  line chart is drawn for a dataset derived from a well known image Lena. The second is for a simple image of a crossroad. The third is for image presenting collection of three toy blocks. The fourth line chart shows the graph of gradient module for GCA ordered pixels of a magnetic resonance (MR) image of a knee.

Most of pixels are ordered in such a manner that the values of grade modules in line charts are descending along vertical axis. But a few pixels form characteristic *quills* - a distinct descending lines. These lines and lines visible in  $gl$  charts will be considered in next section.



**Fig. 1.** Line charts of GCA reordered pixels for four images: a) Lena, b) crossroad, c) blocks, d) knee (MR image)

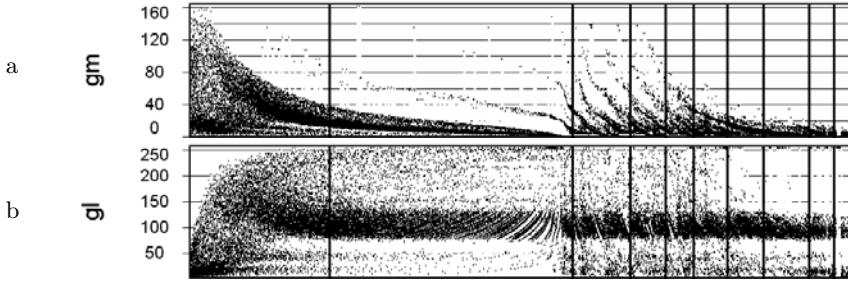


**Fig. 2.** Fragment of a slice of a human brain on MR image

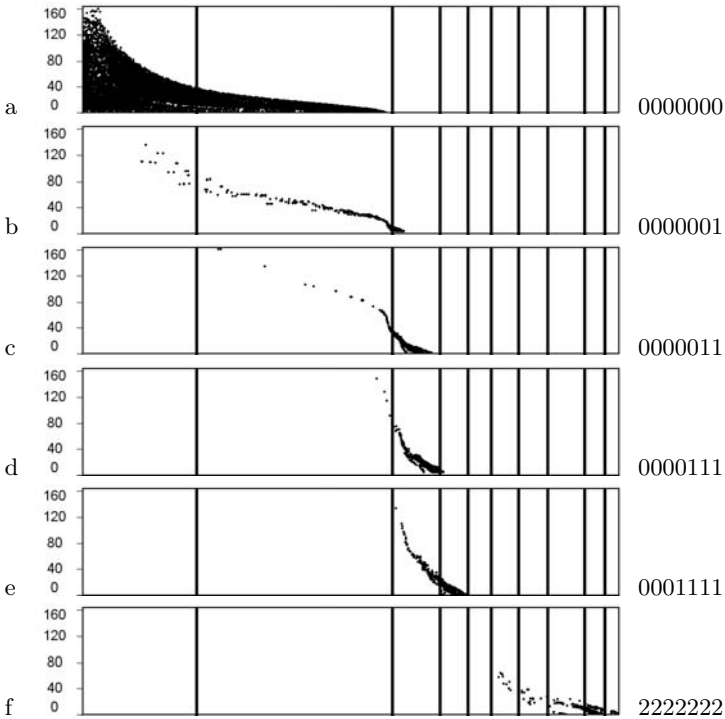
## 4 Experiment Results

In Figure 2 it is shown a fragment of a human brain MR image. The dataset resulting of this image has 61936 rows. Maximum value of gradient module is equal to 161.12,  $b = 0.2$  and  $k = 7$ .

Figure 3 presents line charts of two variables:  $gm$  and  $gl$ , pixels are ordered according to GCA and clustered. In  $gm$  chart it can be seen thin long stripes - *quills*, which descend from left to right. They are narrow at the top of the line chart and widened at the bottom. These quills do not appear in line charts for  $b = 0$ , e.g. for an extreme case when variables  $n_1, \dots, n_k$  are equal to 0 and the procedure GCA can be used only to variables  $gm$  and  $gl$ . It looks like variables



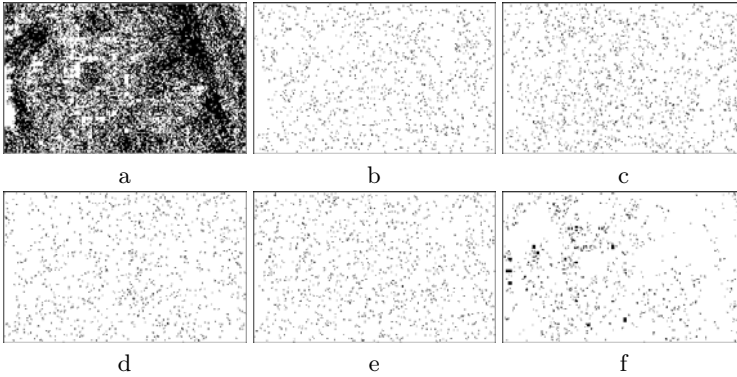
**Fig. 3.** Line charts of gradient module  $gm$  and gray level  $gl$  for GCA ordered pixels of the fragment human brain image from Figure 2



**Fig. 4.** Line charts of  $gm$  for GCA ordered pixels for the image from Figure 2. Marked pixels with selected values of variables  $n_i$ ,  $b = 0.2$ .

$n_i$  (for  $b = 0.2$ ) cause that some pixels in this line chart are moved from their spots (defined by quotient  $gm/gl$  for  $b = 0$ ) to a new regular location. Quills are influenced by variables  $n_1, \dots, n_k$ .





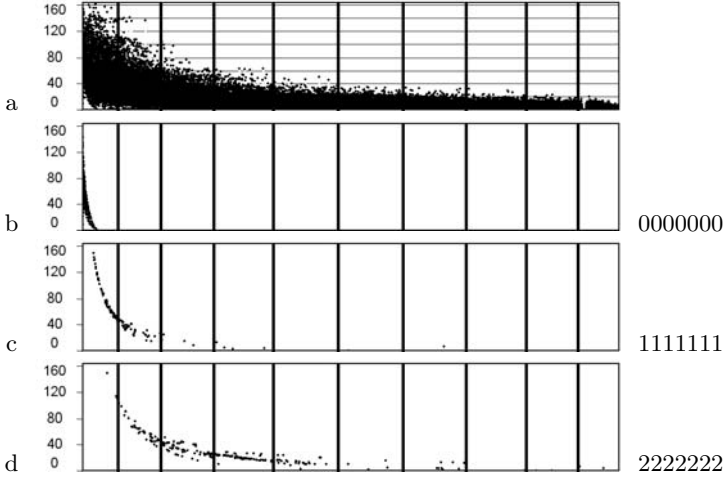
**Fig. 5.** Images visualizing space distribution of the pixels marked in line charts in Figure 4 from a to f

On line charts in Figure 4 are marked selected pixels with fixed values of  $n_i$  only. At the top (Figure 4a) there is the line chart for  $n_i = 0, i = 1, \dots, 7$ . The term *0000000* means that all variables  $n_i$  are equal to 0. On this chart is a quill consisting of pixels with variables  $n_i = 0$  and  $gm$  as follows from the line chart.

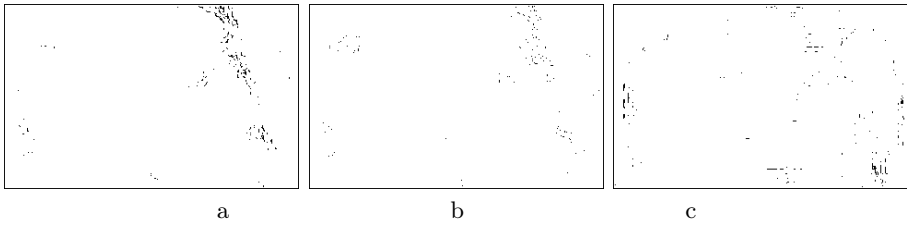
In Figure 4b is the line chart with marked pixels which have value  $n_i = 0$  for  $i = 1, \dots, 6$  and  $n_7 = 1$ . Next line chart (Figure 4c) is for variables' values  $n_i = 0$  for  $i = 1, \dots, 5$  and  $n_i = 1$  for  $i = 5, 6$  and so on. There are 256 different sets of values  $n_1, \dots, n_7$  but only twelve of them is characteristic for over 90% pixels. Some groups of pixels which form separate quills are more distinct, some are small and touching one to another. Figure 5a to 5f show restored spatial positions of pixels marked in line charts from Figure 4a to 4f.

Figure 6 presents line charts of the variable  $gm$  for  $b = 5$ . At the top there is the line chart with all pixels marked (like Figure 3a for  $b = 0, 2$ ), below are the line charts with marked pixels with fixed values of variables  $n_i$ . Pixels with the same values  $n_i$  form quills as for lower values of the parameter  $b$  but these quills are close to each other and are hard distinctive one from another. There are 4718 different sets of values  $n_1, \dots, n_7$ . Figure 7 shows restored spatial positions of pixels marked in line charts from Figure 6b to 6d.

Another kind of structures formed by the GCA reordered pixels are visible in the  $gl$  line chart (Figure 3b). There are thin curves and medium size spots converged near the top of a line which separates second and third cluster. Spots consist of pixels with identical  $n_i$ 's alike quills. Pixels are gathered into curves on a different principle - pixels have a fixed value of variable  $gm$ . Figures 8a and 8b show that there is a number of curves including pixels with the same value of  $gm$  - the pixels are grouped on the basis of their  $n_i$  values. In the line chart for  $gm = 2.5$  (Figure 8b) pixels in the left curve have  $n_i$  pattern equal to *0000000* as can be implied from  $n_1$  line chart in Figure 8c. Next curve, characterized by another  $n_i$  (*(1111111)*), begins with a few pixels near the top of



**Fig. 6.** Line charts of  $gm$  for GCA ordered pixels for the image from Figure 2. Marked pixels with selected values of variables  $n_i$ ,  $b = 5$ .

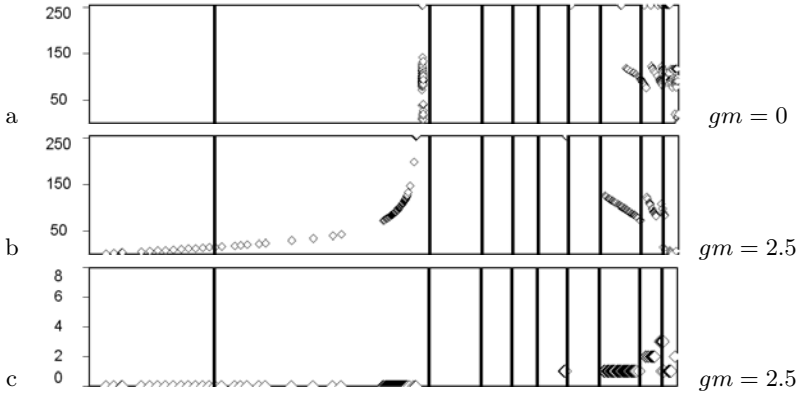


**Fig. 7.** Images visualizing space distribution of the pixels marked in line charts in Figure 6 from b to d

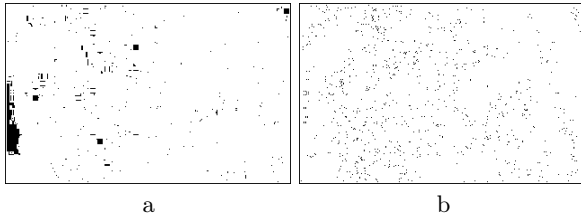
the sixth vertical line, continues through some pixels between seventh and eight vertical line and ends with pixels at the bottom next to right lower corner of the chart. Third and fourth curve with  $n_i = 2$  and  $n_i = 3, i = 1, \dots, 7$  form two remaining small segments of curves. There are four different  $n_i$  patterns among pixels with  $gm = 2.5$ . Another  $gl$  line charts with different values of variable  $gm$  are similar but there are more curve segments. For  $gm = 3.5$  there is 18 different  $n_i$ 's. It is too difficult to show them in one chart and these groups contain fewer pixels.

There are many different  $gm$  values in the dataset. For  $b = 0.2$  there are 3445 different values of  $gm$ , for  $b = 5$  there are 4718.

In Figures 9a and 9b are restored spatial positions of pixels marked in the line charts in Figures 8a and 8b.



**Fig. 8.** Line charts for GCA ordered pixels for the image from Figure 2: a)  $gl$  line chart,  $gm = 0$ , b)  $gl$  line chart,  $gm = 2.6$ , c)  $n_1$  line chart,  $gm = 2.5$



**Fig. 9.** Images visualizing space distribution of the pixels marked in line charts from Figure 8a and b

## 5 Conclusions

Procedure GCA reorders the rows and the columns of multivariate data matrix and the maximum of the positive dependence measure implies the possible maximum regularity of the matrix. Groups of pixels with defined values of variables (such as variables  $n_i$ , variable  $gm$ ) form regular structures with distinct features (quills in  $gm$  charts, curves in  $gl$  charts). These groups of pixels are arranged evenly on charts due to their similarity one to another and a similarity/dissimilarity with other pixels. Huge influence on pixel's characteristic has its neighborhood, i.e. character of changes aroused in surrounding of the pixel. Dynamics of changes of the gray level  $gl$  in the near neighborhood is revealed by the variable  $gm$ . The influence of the neighborhood is strengthened through variables  $n_i$ , since they use gradient modules of the pixels belonging to near neighborhood and it means involving gray level of pixels of farther neighborhood. It seems that there are involved robust constrains. However, more similar are pixels with the same neighborhoods in different positions in the image whereas we intend to reveal structures hidden in the image [5]. It could be performed when stronger

space dependencies will be introduced (taking into consideration coordinates of pixels but without effect of an origin coordinate position).

## References

1. Kowalczyk T, Pleszczyńska E, Ruland F (eds)(2004) *Grade Models and Methods for Data Analysis, With Applications for the Analysis of Data Populations*, Series: *Studies in Fuzziness and Soft Computing*, vol. 151, 477 p., Springer Verlag, Berlin Heidelberg New York
2. <http://gradestat.ipipan.waw.pl/>
3. Książek J, Matyja O, Pleszczyńska E, Wiech M. (eds)(2005) *Analysis of medical and demographical data aided by programme GradeStat*. Institute of Computer Science PAS, the Children's Memorial Health Institute (in Polish)
4. Pleszczyńska E (2007) *Application of grade methods to medical data: new examples*. *Biocybernetics and Biomedical Engineering*. no. 3
5. Grzegorek M (2005) *Image Decomposition by Grade Analysis - an Illustration*. Kurzyński M, Puchała E, Woźniak M, Żołnierek A (eds.) *Computer Recognition Systems. Proceedings of the 4th International Conference on Computer Recognition Systems CORES'05*, May 22-25, Rydzyna Castle. Series: *Advances in Soft Computing*. Springer Verlag: Berlin Heidelberg New York, pp 387-394
6. Grzegorek M (2005) *Decomposition of the fragment of brain image performed using grade data methods with the aid of GradeStat program*. Rutkowski L. (ed.), *Proc. of XIVth Scientific Conference "Biocybernetics and Biomedical Engineering" Poland, Częstochowa, 21-23 Sept*, pp 260-265 (in Polish)
7. Grzegorek M (2007) *Homogeneity of pixel's neighborhoods in gray level images investigated by the Grade Correspondence Analysis*. Kurzyński M, Puchała E, Woźniak M, Żołnierek A (eds) *Computer Recognition Systems. Proceedings of the 5th International Conference on Computer Recognition Systems CORES'07*, Oct 22-25, Wrocław. Series: *Advances in Soft Computing*. Springer Verlag: Berlin Heidelberg New York
8. Gonzalez RC, Woods RE (2002) *Digital Image Processing. Second Edition*. Upper Saddle River, Prentice Hall

---

# A Simple Method of Multiple Camera Calibration for the Joint Top View Projection

Mikhail Mozerov<sup>1</sup>, Ariel Amato<sup>1</sup>, Murad Al Haj<sup>1</sup>, and Jordi Gonzàlez<sup>2</sup>

<sup>1</sup> Computer Vision Center and Department d'Informàtica. Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain  
[mozerov@cvc.uab.es](mailto:mozerov@cvc.uab.es)

<sup>2</sup> Institut de Robòtica i Informàtica Industrial (UPC-CSIC), Edifici U Parc Tecnològic de Barcelona. 08028, Spain

**Summary.** A simple method for multiple camera calibration based on a novel geometric derivation is presented. The main advantage of this method is that it uses only three points in the world coordinate system to achieve the calibration. Rotation matrix and translation vector for each camera coordinate system are obtained via the given distance between the vertices of the marker triangle formed by the three points. Therefore, the different views from the different cameras can be converted into one top view in the world coordinate system. Eventually, the different trajectories traced by certain tracked agents on the floor plane can be obtained from different viewpoints and can be matched in a joint scene plane.

## 1 Introduction

Camera calibration techniques play an important role in the different applications of computer vision. The objective of the calibration process is to obtain both the intrinsic and the extrinsic camera parameters. The intrinsic parameters are decided by the inner geometry and the optical characteristics of the camera; these include focal ratio and radial distortion factor. The extrinsic parameters reflect the relationship between the image plane and the world plane; these include rotation matrix  $\mathbf{R}$  and translation vector  $\mathbf{T}$ . However, the importance of each of these parameters, intrinsic or extrinsic, depends on the problem to be solved and on the camera model. For example, a new generation of digital cameras provides rectified image sequences; therefore, the obtained images are not affected by the radial distortion of the camera. Most traditional camera calibration techniques require specific knowledge about the geometric characteristics of the referenced object, such as Direct Linear Transformation (DLT)[1], which solves the perspective matrix linearly; similar methods include Tsais[2], and Zhangs [3]. Plane based methods[4]-[5], use the same DLT paradigm, but show more flexibility. These methods use multiple view approach for each camera and are not very useful for our problem. The main goal of this work is to use calibration in order to convert the image planes obtained from multiple camera views into one top view of an inspected scene, as shown in Fig. 2. This allows us to interpret certain actions, after extracting them from the scene, in a chosen world coordinate system. The similar approach is used in the paper of Lee et al.[6]. This can be done

by performing trajectory matching of the tracked foot points of the moving agent. Many works use these approach to handle uncertainty of one point view[7]-[8]. In such a scenario, the most important aspect is to achieve sufficient accuracy in the ground plan which contains the marker triangle (the floor plane on which the agent is walking). Our experiments show that in this situation, the focal ratio does not play an important role and the lens focus scope ( $\pm 5\%$ ) produces minimal distortion in the ground plan measurements. Furthermore, in the extreme case where the image plane and the world plane coplanar, the accuracy of the ground plane measurements do not depend on the focal ratio. Of course, the focal ratio is essential in determining the distance between the camera and the scene or in measuring the heights of objects in the scene; however, in these case, the focal length can be obtained a priori, either through a simple geometric approximation or from the datasheet of the camera and lens. The DLT methods require more than three points in the world coordinate system, for example, the Tsais calibration technique requires five points. Three point problem usually is considered as theoretical problem without implementing in real experiments[9]-[10]. Our main contribution is to propose calibration method that uses only three world coordinate points, eventually calibration process can be considered as two independent and intuitively clear parts with simple geometrical interpretation; the first part is obtaining values of marker triangle points in a camera coordinate system while the second is deriving  $\mathbf{R}$  and  $\mathbf{T}$ . This paper is organized as follows. Section 2 presents our camera model while section 3 discusses our calibration process. Section 4 describes inverse perspective mapping. Section 5 shows our experimental results. Concluding remarks are made in section 6.

## 2 Camera Model

As shown in Fig. 2 pinhole camera model is used. We suppose that the focal ratio parameter is known or predetermined. The right-hand camera coordinate system  $\mathbf{p}_0, \mathbf{X}_c, \mathbf{Y}_c, \mathbf{Z}_c$  is defined as origin  $\mathbf{p}_0$ , which coincide with optic center of camera,  $\mathbf{Z}_c$  axis coincident with optical axis and has invers direction, which means that all visible point including projection points have depth values strongly less than zero. A scene point  $\mathbf{P}$  can be represented in the camera coordinate system as  $\mathbf{P}_n = [X_n, Y_n, Z_n]^t$  and in the world coordinate system ( $\mathbf{O}w, \mathbf{X}_w, \mathbf{Y}_w, \mathbf{Z}_w$ ) as  $\hat{\mathbf{P}}_n = [\hat{X}_n, \hat{Y}_n, \hat{Z}_n]^t$ . The projection of any world point onto image plane, which is parallel to the camera plane, is denoted by lowercase as  $\mathbf{p}_n = [x_n, y_n, -f]^t$ , where  $f$  is the camera focal length. The relationship between these two coordinate values is as follow.

$$\mathbf{P}_n = \frac{Z_n}{f} \mathbf{p}_n \quad (1)$$

As we work with the raster images, it is useful also to consider the raster representation of the image plane points  $\mathbf{m}_n = [i_n, j_n, -f_{pix}]^t$ ,  $i \in [-I, I]$ ,  $j \in [-J, J]$ , where  $f_{pix}$  is the focal length expressed in the raster pixels,  $I$  and  $J$  are the half size of the image matrix in  $X$  and  $Y$  directions respectively. If to denote

the distance between two neighbor raster pixels as  $\mu = CCD_x/2I$ , where  $CCD_x$  is the physical size of the camera CCD matrix in  $X$  direction, then, the focal length in pixels is  $f_{pix} = f/\mu$ . Let us denote the inverse of the intrinsic constant  $f_{pix}^{-1}$  as  $\varphi$ , and now we are prepared to express an image plane point via raster representation:

$$\mathbf{p}_n = \mu \mathbf{m}_n = f \mathbf{u}_n \tag{2}$$

where  $\mathbf{u}_n = \varphi \mathbf{m}_n = [\varphi i_n, \varphi j_n, -1]^t$  is a convenient raster representation of a point in the image plane.

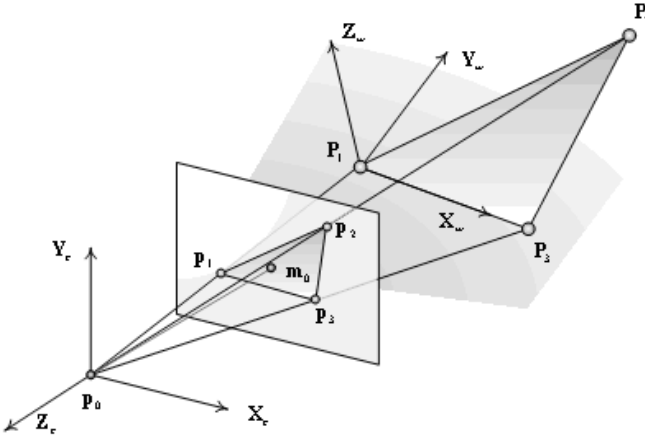


Fig. 1. Pinhole camera model for calibration with three world coordinate points

To obtain focal ratio of our cameras we use a simple geometrical approach with just two points: it is assumed that two world coordinate points  $\mathbf{P}_1, \mathbf{P}_2$  have two given projection on the image plane  $\mathbf{p}_1, \mathbf{p}_2$  as it is shown in Fig. 1. We also suppose that the distance between the two world points  $d_{12} = |\mathbf{P}_2 - \mathbf{P}_1|$  is known and distances between optical center  $\mathbf{p}_0$  and given points  $\mathbf{P}_1, \mathbf{P}_2$  are measured (for example using laser distancemeter as it is in our experiments). Now, the angle between unknown vectors  $\mathbf{p}_1, \mathbf{p}_2$  can be calculated using cosine theorem, and finally  $\cos(\mathbf{P}_1, \mathbf{P}_2) = \cos(\mathbf{p}_1, \mathbf{p}_2)$ . Then the focal ratio is the a simple solution of this equality.

### 3 Calibration Algorithm

Now, we assume that the intrinsic constant  $\varphi$  is given or predetermined. Note, that in this case only the depth parameter  $Z_n$  is needed to completely describe all the visible points in the world coordinate system of a camera (see Eq.( 1- 2). On the other hand, three points of the world coordinate system (if they do not belong to the same line) are enough to determine extrinsic coordinate system. So, the calibration process can be divided into two independent parts. First, we

calculate values of marker triangle points in a camera coordinate system and then obtain extrinsic parameters  $\mathbf{R}$  and  $\mathbf{T}$ . Let us denote the distances between three point in the marker triangle  $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$  and scalar product matrix of its projection points as  $d_{kl} = |\mathbf{P}_k - \mathbf{P}_l|$  and  $\sigma_{kl} = \mathbf{u}_k \mathbf{u}_l$  with  $k, l=1, 2, 3$ . Then the three perspective points problem can be defined as follow: find three desired values  $Z_1, Z_2, Z_3$  that satisfy the system of three equations

$$Z_k^2 - 2Z_k Z_l a_{kl} + Z_l^2 b_{kl} - c_{kl} = 0 \tag{3}$$

where  $a_{kl} = \sigma_{kl} / \sigma_{kk}$ ,  $b_{kl} = \sigma_{ll} / \sigma_{kk}$ ,  $c_{kl} = d_{kl} / \sigma_{kk}$ .

The system of these nonlinear equations can be solved in two ways. One way leads to a biquadratic closed form, which is a big problem itself. We propose another way with a fast iterative algorithm that provides computer level accuracy. Let us suppose that  $0 > Z_3 \geq Z_1 \geq Z_2$ . In this case the system has just one solution and the values of  $Z_3, Z_1$  can be calculated via value  $Z_2$

$$Z_k = Z_2 a_{k2} + \sqrt{Z_2^2 (a_{k2}^2 - b_{k2}) - c_{k2}} = 0, \quad k = 1, 3 \tag{4}$$

We remember that depth values of the marker triangle points satisfy  $0 > Z_3 \geq Z_1 \geq Z_2$  and the expression under square root in Eq.( 4) must not be negative. These constraints provide us with the limits of  $Z_2$  variable domain. Then substituting Eq.( 4) in ( 3) we have one equation to solve

$$(Z_1(Z_2))^2 - 2Z_1(Z_2)Z_3(Z_2)a_{13} + Z_l^2 b_{13} - c_{13} = 0 \tag{5}$$

This equation with given order constrain  $0 > Z_3 \geq Z_1 \geq Z_2$  has only one solution within the calculated segment  $[\min(Z_2), \max(Z_2)]$ . In such a case, the solution can be obtained iteratively using dichotomic division algorithm of the desired variable domain. The double float precision computer accuracy can be achieved with less than 50 steps. Now we are able to derive rotation matrix and translation vector that transform the new world coordinates of a point into the camera world coordinates

$$\mathbf{P}_n = \mathbf{R} \hat{\mathbf{P}}_n + \mathbf{T} \tag{6}$$

To obtain the transformation parameter, first, it is necessary to define new world coordinate system. Let us put the origin of our world coordinate system to one of the marker triangle vertices  $\mathbf{O}_w = \mathbf{P}_1$ , and take one of the triangle leg as the  $\mathbf{X}$  coordinate axis. It is reasonable to put our marker triangle into the new coordinate system plane with  $Z=0$ . Then, the new world coordinate system can be defined by its basis

$$\mathbf{X}_w = \frac{(\mathbf{P}_3 - \mathbf{P}_1)}{|\mathbf{P}_3 - \mathbf{P}_1|}; \quad \mathbf{Z}_w = \frac{\mathbf{X}_w \times (\mathbf{P}_2 - \mathbf{P}_1)}{|\mathbf{X}_w \times (\mathbf{P}_2 - \mathbf{P}_1)|}; \quad \mathbf{Y}_w = \mathbf{Z}_w \times \mathbf{X}_w; \tag{7}$$

Now, rotation matrix can be represented as a simple combination of the basis vectors and translation vector is equal to origin vector

$$\mathbf{R} = [\mathbf{X}_w \mathbf{Y}_w \mathbf{Z}_w]; \quad \mathbf{T} = \mathbf{P}_1; \tag{8}$$



## 4 Inverse Perspective Mapping

One of the implementations of the multiple camera calibration is inverse perspective mapping. Once your tracking algorithm localizes the pixel of interest  $(i,j)$  in the image of one camera (in our case it is a foot point of an agent) you have to project this pixel onto joint plane in the world coordinate system for the matching with another point obtained from the image of the second camera. This projection process is referred to as inverse perspective mapping. In other words, we need to derive a function  $\hat{\mathbf{P}}(i,j)$  that project arbitrary pixel with indexes  $(i,j)$  into the real world plane. To derive the desired function, first, we need to obtain the value of the optical center point in the joint world coordinate system

$$\hat{\mathbf{p}}_0 = \mathbf{R}^t (\mathbf{p}_0 - \mathbf{T}) = -\mathbf{R}^t \mathbf{T} \quad (9)$$

Let us denote the pure rotation of the image plane vectors as

$$\check{\mathbf{u}}_{ij} = \mathbf{R}^t \mathbf{u}_{ij} = \mathbf{R}^t [\varphi i, \varphi j, -1]^t \quad (10)$$

then, the desired inverse perspective mapping function is

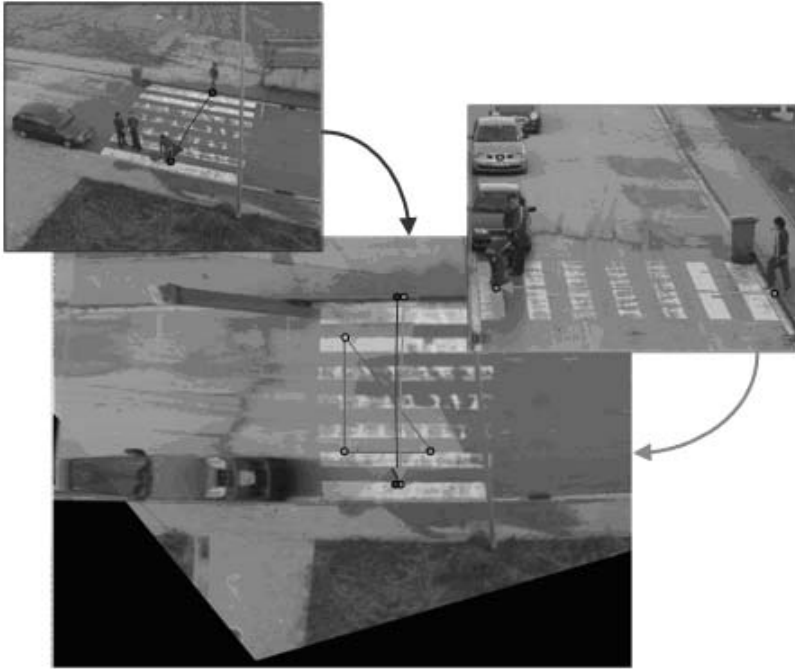
$$\hat{\mathbf{P}}_{ij} = \hat{\mathbf{p}}_0 - \check{\mathbf{u}}_{ij} \frac{\hat{z}_0}{\check{z}_{ij}} \quad (11)$$

If the localization algorithm is working properly and pixels of interest  $(i,j)$  in the image of one camera and  $(i',j')$  in the image of other camera belong to the same agent the distance between these two points  $|\hat{\mathbf{P}}(ij) - \hat{\mathbf{P}}(i'j')|$  in the joint world coordinate plane might be zero. Of course, the excellent accuracy of the method can be flawed by the real world ground curvature, and to check it we performed experiments with real image sequences.

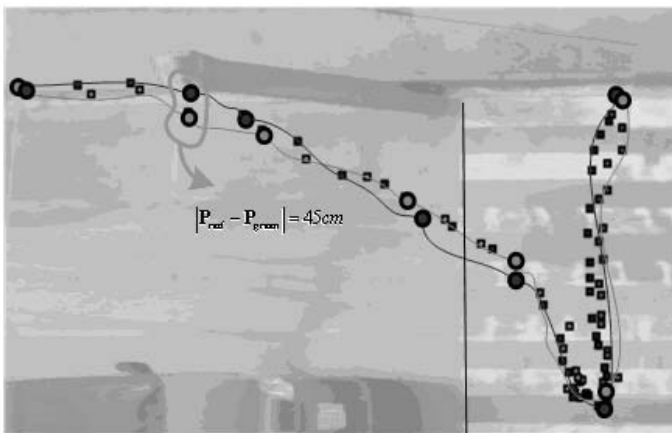
## 5 Computer Experiments

In our experiments we use the real data sequences of two view point. In general the goal is to compare trajectories form different points of view in one joint plane. The marker triangle (Egyptian triangle with 3x4x5 meter legs) is formed using the corner points of colored square markers Fig. 2. To check the accuracy of the method we map two images in one joint top view. The green border of the in Fig. 2 shows the overlapping effect of the two inverse mapping. We can see that visual features (zebra crossing lines e. g.) coincide on the arbitrary drawn fringe. The inverse perspective mapping also performed for two trajectories taken from different cameras and belongs to the same tracked agent (the green colored line from the first camera and red colored line from the second).

We estimate the distances in the joint ground plan and maximum error value is 45 cm. Note, that this error is not a result of inaccurate calibration or inverse mapping, but mostly due to inaccuracy of the foot point localization in the image. The no planarity of the scene affects on the curvature of zebra crossing



**Fig. 2.** Projection of two multiple camera views onto the top view of the inspected scene



**Fig. 3.** The result of two trajectory matching

relative to strait black line in Fig. 3, but this effect produces error measurements less than 10 cm in the field of interest. So, we can conclude that the proposed calibration method has a sufficient accuracy for handling multiple cameras views.

## 6 Conclusion

We propose simple calibration method, which is sufficient for multiple camera handling. Trajectories of the foot points of the tracked agents are usually not very accurate due to intrinsic property of segmentation and tracking algorithms. The experimental results show that the accuracy of the trajectories matching with proposed calibration and geometrical mapping technique highly overpass tracking approximation.

## Acknowledgements

This work is supported by EC grants IST-027110 for the HERMES project and IST-045547 for the VIDi-video project, and by the Spanish MEC under projects TIN2006-14606 and DPI-2004-5414. M. Mozerov and Jordi Gonzàlez also acknowledge the support of a Ramon y Cajal and Juan de la Cierva Postdoctoral fellowships from the Spanish MEC, respectively.

## References

1. Y. I. Abdel-Aziz, and H.M. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry", *Proc. of the ASP Symposium on Close-Range Potogrammetry*, pp. 1–18, 1971.
2. Tsai, R., "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, *IEEE J. Robot. Autom.*, Vol. 3 pp. 323–344, 1987.
3. Zhang, Z, "A flexible new technique for camera calibration, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22 pp. 1330–1334, 2000.
4. Sturm, P., and Maybank, S., "On plane-based camera calibration: A general algorithm, singularities, applications, *IEEE CVPR*, pp. 432–437, 1999.
5. Lucchese, L., "Geometric calibration of digital cameras through multiview rectification, *Image Vis. Comput*, Vol. 23 pp. 517–539, 2005.
6. Lee, L., Romano, R., Wang, L., and Stein, G., "Monitoring activities from multiple video streams: Establishing a common coordinate frame, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22 pp. 758–767, 2000.
7. Cai, Q., and Aggarwal, J. K., "Human motion in structured environments using a distributed camera system, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 21 pp. 1241–1247, 1999.
8. Chang, T.-H., and Gong, S., "Tracking multiple people with a multi-camera system, *IEEE Workshop Multi-Object Tracking*, pp. 19–26, 2001.
9. Haralick, R., Lee, C., Ottenberg, K., Nolle, M., "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem, *International Journal of Computer Vision*, Vol. 13 pp. 331–356 1994.
10. Nister, D., "A minimal solution to the generalized 3-point pose problem. On plane-based camera calibration A general algorithm, singularities, applications, *IEEE CVPR Volume 1* pp. I560–I567 2004.

---

# Spiral Scanning of Faces to Obtain Complete 3D View Representation of Monotonous Polyhedra

Wojciech S. Mokrzycki<sup>1</sup> and Andrzej Salamończyk<sup>2</sup>

<sup>1</sup> Faculty of Mathematics and Informatics, University of Warmia and Mazury,  
Zolnierska 47, Olsztyn, Poland  
mokrzycki@matman.uwm.edu.pl

<sup>2</sup> Institute of Informatics, University of Podlasie, Sienkiewicza 51, Siedlce, Poland  
andrzej@ii.ap.siedlce.pl

**Summary.** In this paper we propose a new method of generation 3D multiview representation of monotonous polyhedrons on the base of their  $B_{rep}$ . The algorithm acts according to viewing sphere with perspective projection concept (called *K-M view space model*), scanning polyhedron faces in spiral way. Obtained models are used for visual identification based on them and a scene depth map. The results of algorithm testing are presented.

## 1 Introduction

3D multiview exact representation of polyhedron is mainly used for object visual identification. It is based on multiview models and a scene depth map. The following steps, which constitute an element of strategy are usually taken in order to recognize a given system:

1. Determination of recognizable object types.
2. Definition of identification task.
3. Generation of viewing models for each object that should be identified.
4. Creation of database with all views of all models.
5. Acquisition of scene space data and visual data.
6. Isolation of scene elements and transformation of those elements to model structures stored in the database.
7. Identification of objects by comparing them with database models.

Generation of viewing models for each object that should be identified is one of the first and main step for acting identification system.

Methods of generating 3D multiview representation of polyhedron for object visual identification are described in several papers, e.g. [2] ÷ [4], [6] ÷ [8], [10] ÷ [13], [16, 17]. Some of them - concerning models for convex polyhedrons only and called *iterative methods* (e.g., presented in [3]), can be described by a series of steps:

- First, central views corresponding to object features chosen for identification are generated.

- Next, *single-view areas* are calculated on the *view sphere*. They correspond to views generated earlier.
- Then, it is checked if the whole viewing sphere is covered with single-view areas.
  - If this is completed, generation of viewing representation is finished.
  - If not, then additional views corresponding to uncovered areas of viewing sphere are generated and it is checked again if the process is completed.

Complete viewing sphere cover with single-view areas means that generated representation is complete. Completeness of the view representation means that all different views (aspects) of the object are determined for each point on the view of the sphere and a view is defined.

Papers [6, 8] concern methods called *noniterative*, better as faster. Complete representation is obtained by **covering the viewing sphere precisely with single-view areas without loop, but in spiral way and controlling „edge” register** (of not covered area). **When the register is set to „empty” the generation of representation is completed.** The representation used is complete, which results from the generation method used. However, to achieve complete representation you must calculate single-view areas on viewing sphere and carry them out in a given order. Without single-view areas it is not possible to find a complete set of views. All described methods produce convex polyhedron views only.

Next group of methods in [7, 10, 12] **does not use any single-view areas**, so are faster than the previous ones. They are based on other principle which consists of the following steps: **A complementary cone** (s.c. *scanning cone*) rotates around faces normals and records every *visual event* to obtain all views. This event occurs as a result of a new normal versor entering the scanning cone or by disappearance of a versor. The outcome of such a routine is a set of vectors faces that can be seen from the view sphere. However, they were tested for convex polyhedrons only, [12].

In [13] authors presented a method (the algorithm and its implementation) based on following idea: **complementary cone** to obtain views, rotate around faces (no around faces normals) as in Fig. 1b and record every *visual event*. The followed algorithm generates 3D exact views for (not only convex polyhedrons but also) monotonous polyhedrons (monotonous polyhedrons are a class of non-convex polyhedrons). However this implementation does not include shadows in the views. The presented method in this work (algorithm and its implementation) concerns views with shadows. Implementation presented in [16] just includes shadows in the views also.

In this paper **we present entirely new method**, which will be suitable (as we believe) to generate not only 3D multiview polyhedrons representation. It consists in moving scanning cone along spiral way of viewpoint around given normal.

## 2 Research Assumptions

We concentrate on developing our latest method and algorithm [13], generating 3D exact view representation for monotonous polyhedrons taking shades into consideration. For representation generation we use viewing sphere with perspective projection (*K-M view model* [3]). For the following conditions have to be met:

1. Models are accurate - every model is equivalent to  $B_{rep}$  model.
2. Models are viewing models - it is possible to identify object from any view.

We consider polyhedrons that are non transparent and monotonous. As we use a viewing sphere with central projection as a projection space, we allow simple view standardization.

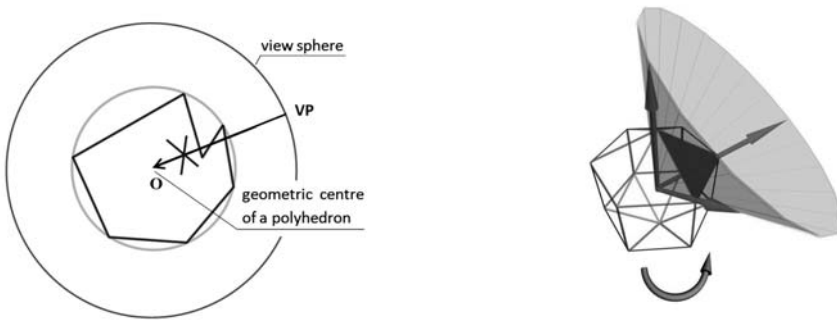


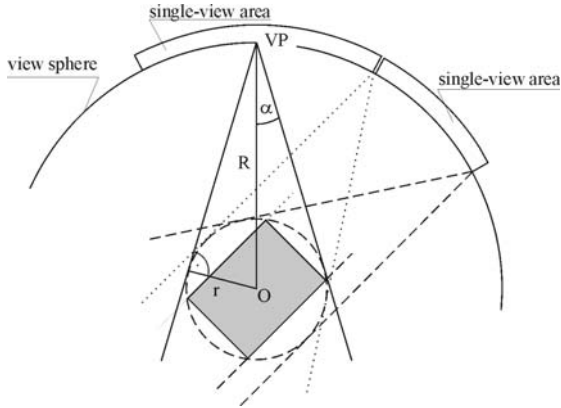
Fig. 1. Monotonous polyhedron and scanning face surrounding

## 3 View Generation Space - Basic Concepts

Let object be a monotonous, non transparent polyhedron without holes or pits. Consider its faces as feature areas, those areas will be used as a foundation for accurate multiview 3D model determining. This model is a set of accurate views, acquired through perspective projection from viewing sphere, according to the model from [3]. This model is the best for 3D scene data acquisition as it gives identification system reliability.

The concept of generating 3D multiview representation based on assumed generation space model is as follows:

- Circumscribe a sphere on a polyhedron. The sphere is small (radius  $r$ ) and its center is at the polyhedron center.
- On this sphere place a space view cone ( $vc$ ) with angle of flare  $2\alpha$ . This is the **viewing cone**. The vertex of this cone is a model **viewing point**  $VP$ . The distance between polyhedron center and model viewing point -  $R$ . Viewing axis always goes through sphere (and the same object) center.



**Fig. 2.** Concept of view sphere and „single view areas”, [3]

- Unconstrained movement of the cone vertex, where the cone is tangent to the small sphere creates a large sphere with radius  $R$ . This sphere is called **viewing sphere** (Fig. 2). Each object has its own viewing sphere, the same for all views of this particular object.
- Generate views, taking into account only object features selected for identification i.e. faces. Faces visible in the viewing cone create a **view**, external edges from this view create **view contour**.

The dependency between  $r$  and polyhedron vertices coordinates  $(Xvi, Yvi, Zvi)$  and:  $R, \alpha, r$  and angle of view cone  $vc$  flare are:

$$r = \max_{i=1, \dots, k} \sqrt{Xvi^2 + Yvi^2 + Zvi^2}, \quad R \geq \frac{r}{\sin \alpha}, \quad \angle(vc) = 2\alpha.$$

Changing one view to the other is a **visual event**. This event occurs as a result of point  $VP$  movement and is manifested by appearance of a new feature in a view, disappearance of a feature or both (Fig. 2).

**Complementary cone** ( $cc$ ) is a cone defined by current viewing axis (it's collinear with its height) and has an opposite direction of flare to the viewing cone. It intersects viewing cone with angle  $\pi/2$ , so its angle of flare is:  $\angle(cc) = \pi - 2\alpha$ .

**Vector Representation of Polyhedron**  $V_{rep}$  is a set of polyhedron faces normals translated to point  $(0, 0, 0)$  (Gauss map) [7].

View is created by faces that are visible in the viewing cone at a certain viewing point  $VP$  position. External edges of a view create a viewing contour.

**Natural Representation of Polyhedron** ( $N_{rep}$ ) is a set of pyramids, which have common top in the geometric center of the polyhedron and whose bases are on the polyhedron faces.

**Monotonous Polyhedrons**, [9], is a class of nonconvex polyhedrons in which pyramids of  $N_{rep}$  do not pierce.

**Edge Vector** is a vector that has starts in polyhedron center and ends in one of polyhedron vertex. It is collinear with some edges of pyramids from  $N_{rep}$ .

## 4 Mathematical Description of Spiral Motion

In this section we describe motion of view point in mathematical way and prove that the algorithm generates complete set of views (*exact viewing representation*). We start from determining the resolution of scanning:

$$\Delta\varphi = \min_{i,j} \{\angle(N_i, N_j) - 2\beta \neq 0\} \quad (1)$$

where  $\angle(N_i, N_j) \in [0, 2\pi)$  stands for angle between normals of two faces and  $\beta$  denotes 1/2 of flare of complementary cone ( $\beta = \pi/2 - \alpha$ ). First, we perform spiral motion of view point around given normal. For simplicity reason we described the movement around positive „z” axis and it is described in parametric equation in spherical coordinates.

$$\begin{cases} \theta = k\sqrt{t} \\ \phi = \sqrt{t} \end{cases} \quad (2)$$

where  $t \in [0, \beta^2)$ . The zenith angle is denoted by  $\phi$  is a zenith angle and  $\theta$  stands for an azimuth angle. The factor  $k$  determines the angle between successive coin of spiral(it is equal to  $2\pi/k$ ). We put  $k = \left\lceil \frac{2\pi}{\Delta\varphi} \right\rceil + 1$  where  $\Delta\varphi$  is determined by the equation 1. The square root in above equation is responsible for traveling equal distance (approximate) for constant change of parameter  $t$ .

After the spiral motion we perform full rotation the complementary cone around face's normal.

$$\begin{cases} \theta = t \\ \phi = \beta \end{cases} \quad (3)$$

where  $t \in [k\beta, k\beta + 2\pi)$ .

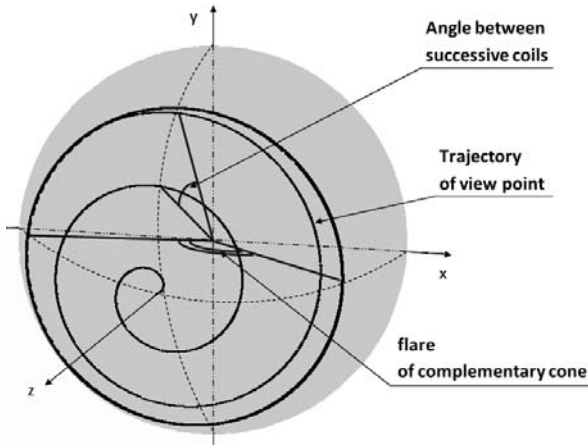
In Cartesian coordinates we describe this motion as follows:

$$\begin{cases} x = \cos \theta \sin \phi \\ y = \sin \theta \sin \phi \\ z = \cos \phi \end{cases} \quad (4)$$

Trajectory of scanning around normal is show on fig. 3. We register views and visual event during this scanning.

**Theorem 1.** *The set of views obtained by the algorithm is complete (there is no additional views on view sphere).*





**Fig. 3.** Trajectory of scannig around normal in spiral way

*Proof.* (on the contrary) Assume that there is at least one view on view sphere which is not obtained by the algorithm. Then there exists view point (let us denote them by  $VP'$ ) on view sphere, which gives this view. We have at least one normal in  $cc$  (denoted by  $N_{in}$ ) and at least one normal outside the  $cc$  (denoted by  $N_{out}$ ). If  $VP'$  lies on the trajectory of scanning it means that we already obtained this view and we have contradiction with the assumption. If  $VP'$  lies outside the trajectory we move slightly  $VP'$  in any direction in such way that the angle between  $VP'$  and every normals lies outside  $cc$  remains less then  $\beta$ . Such a movement do not change the view because:

- None of normals will not enter the  $cc$ ,
- None of normals will not leave the  $cc$ .

If the second case occurs it means that  $VP'$  cross the trajectory of scanning around this normal (the second phase of scanning, see equation 3). Therefore we should deal with the firs case. Whereas this movement we cross the trajectory we prove the theorem. If not we move in opposite direction with maximum angle equals  $\Delta\varphi$ . None of normal lies outside  $cc$  do not enter  $cc$ . It follows from the definition of  $\Delta\varphi$  (see equation 1). Because we can move  $VP'$  in any direction with the angle equal to  $\Delta\varphi$  in such way that normals lies outside do not enter  $cc$  we surely cross the trajectory of scanning. It follows from the method of determination the factor  $k$ . We have shown that is impossible to obtain view not generated by our algorithm because all potential views are generated (normal of face in  $cc$  is necessary condition of visibility of this face) and the proof is finished.

## 5 The Results and Conclusion

The algorithm works - as one can see on figs. 4, 5, and we are testing it on other monotonous polyhedrons. We expect it occurs to be very robust algorithm

not for generation of 3D multiview representation of (monotonous) polyhedrons only, but also for solids with curved faces, what we plane to research.

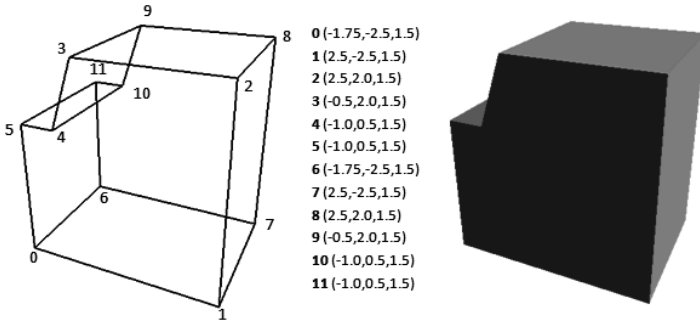


Fig. 4. Monotonous octahedron (12 vertices, 18 edges) used for testing spiral scanning algorithm

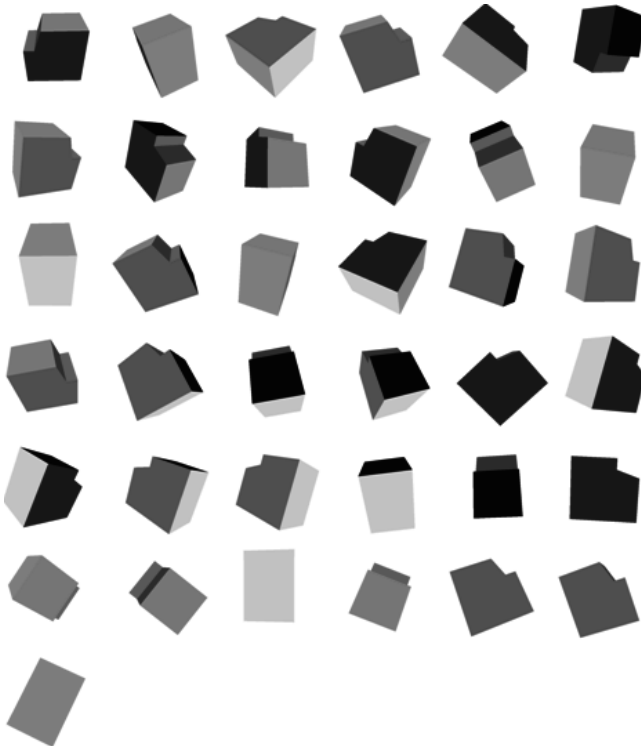


Fig. 5. Obtained 37 views,  $\alpha = 22^\circ$

## References

1. Bern M., Dobkin D., Eppstein D., Grossman R.: Visibility with a moving point of view. Proc. 23<sup>th</sup> ACM Symp. on Theory of Computing, 1(3).
2. Arbel T., Ferrie F.P. 1996: Informative views and sequential recognition. Proc. ECCV'96, Cambridge, UK, April, 469-481.
3. Dabkowska M., Mokrzycki W.S. 1997: Multi-view models of convex polyhedron. MG&V, 6(4), 419-450.
4. Dabkowska M., Mokrzycki W.S. 1998: A new view model of convex polyhedron with feature dependent view. MG&V, 7(1/2), (Proc. GKPO'98, Borki, Poland, 18-22 May), 325-334.
5. Stamos I., Allen P.K.: Interactive sensor planning. Computer Vision and Pattern Recognition. Proc. IEEE Comp. Soc. Conf., 23-25 Jun., 489-494.
6. Kowalczyk M., Mokrzycki W.S. 2003: Obtaining complete 2 1/2D view representation of polyhedron using concept of seedling single-view area. CV&IU 91, 208-301.
7. Frydler M., Mokrzycki W.S.: 2 1/2D view models of nonconvex polyhedron on view sphere with perspective. Proc. Advance Computer Systems'03, Midzysdroje, 22-24 X.
8. Kowalczyk M., Mokrzycki W.S.: Methods of generation 3D exact views of convex polyhedron for visual identification. Part II: Noniterative methods, implementation and tests results. MG&V, 12(4), 435-452.
9. Mokrzycki W.S.: 3W modele widokowe niewypukłych wielocianów na sferze widokowej z perspektyw. Raport Prace IPI PAN, nr 962.
10. Frydler M., Mokrzycki W.S.: New, fast algorithm of 3D multiview polyhedron representation generation on view sphere with perspective. Proc. CCV(ACV)'04, Prague, May, 15-16.
11. Kowalczyk M., Mokrzycki W.S.: Concept of 3D view representation generation of nonconvex polyhedron with view sphere partitioning on single view areas. Proc. 4th Int. Conf. on Computer Recognition Systems CORES'05, Rydzyna, May 22-25, Springer, 411-418.
12. Frydler M., Mokrzycki W.S.: Scanning faces surroundings - new concept in 3D exact multiview of nonconvex polyhedron generation. Proc. 4th Int. Conf. on Computer Recognition Systems CORES'05, Rydzyna, May 22-25, Springer.
13. Mokrzycki W.S., Salamończyk A.: Generating 3D multiview polyhedron representation by scanning faces surrounding. ICCVG2006, Warsaw, 25÷27 Sept.
14. Cohen D., Chrystanthou Y., Silva C.T., Durant F.: A survey of visibility for walkthrough applications. <http://people.csail.mit.edu/fredo/PUBLI>
15. Ghali S.: A geometric framework for computer graphics addressing modeling, visibility, and shadows. [www.cs.ualberta.ca/~ghala/thesis](http://www.cs.ualberta.ca/~ghala/thesis)
16. Mokrzycki W.S., Salamończyk A.: Using shaded views in 3D multiview representation of monotonous polyhedron. Accepted on CISIM 2007, Elk, 29 June - 1 July.
17. Mokrzycki W.S., Salamończyk A.: New results in generation of accurate 3D multiview representation of monotonous polyhedrons - Cores 2007. Contributed on Cores 2007.

---

# Face Detection in Color Images Using Primitive Shape Features

Murad Al Haj<sup>1</sup>, Ariel Amato<sup>1</sup>, Xavi Roca<sup>1</sup>, and Jordi González<sup>2</sup>

<sup>1</sup> Computer Vision Center and Department d'Informàtica. Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain

malhaj@cvc.uab.es

<sup>2</sup> Institut de Robòtica i Informàtica Industrial (UPC-CSIC), Edifici U Parc Tecnològic de Barcelona. 08028, Spain

**Summary.** Face detection is a primary step in many applications such as face recognition, video surveillance, human computer interface, and expression recognition. Many existing detection techniques suffer under scale variation, pose variation (frontal vs. profile), illumination changes, and complex backgrounds. In this paper, we present a robust and efficient method for face detection in color images. Skin color segmentation and edge detection are employed to separate all non-face regions from the candidate faces. Primitive shape features are then used to decide which of the candidate regions actually correspond to a face. The advantage of this method is its ability to achieve a high detection rate under varying conditions (pose, scale,...) with low computational cost.

## 1 Introduction

Detecting faces in images is essential in many applications of computer vision. Of these applications, facial expression analysis and human computer interaction have attracted much interest lately. The main challenges facing any face detection system include: pose variation, presence or absence of structural components (beards, mustaches, and glasses), facial expressions, occlusion, image orientation, and imaging conditions. Many techniques exist for solving these problems and trying to achieve a robust detection. A survey for face detection is presented, [1], where the different methods are classified into four, sometimes overlapping, categories: knowledge-based methods, feature invariant approaches, template matching methods, and appearance-based methods. Viola and Jones, [2], have introduced a rapid and robust object detection using Haar-like features and Adaboost and they applied it to face detection. This method has been gaining popularity among researchers. However, the problem with this method is that it is very sensitive to the face pose. In other words, it is very difficult to detect frontal faces and profile faces with the same cascade of weak classifiers. Therefore, two cascades, one for frontal and another for profile, need to be trained for proper classification; and even with these two cascades, some faces whose pose is between frontal and profile, 45 degree for example, can't be detected. Some modifications have been proposed to make this approach more robust for

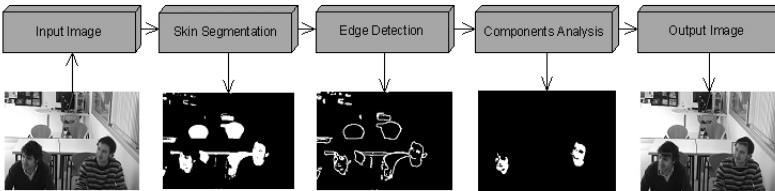
multi-view [3]. However, some researchers have returned lately to the use of color as a robust cue for face detection, [4] and [5], where the proposed methods are based on segmenting the image using human skin color. A method that detects face using the spatial arrangement of skin patches is available in [6].

In this paper, a novel face detection method based on skin color segmentation is presented. The main contributions of this approach can be summarized in proposing an efficient skin color model for image segmentation and using simple primitive shape features to achieve robust detection under varying poses and complex backgrounds.

This paper is organized as follows. Section 2 provides an overview of the system. Section 3 introduces our proposed skin color model while section 4 presents the face localization method. In section 5, experimental results are shown. Concluding remarks are discussed in section 6.

## 2 System Overview

The complete system is shown in Fig. 1. It starts by segmenting the image into regions that contain possible face candidates, while those that do not contain a face object are dropped. This segmentation helps accelerate the detection process. Later, an edge detection filter is applied (i) to separate the candidate faces from any background component that has a color similar to the skin, and (ii) to disconnect foreground components, such as two touching faces, or a hand on the cheek. Next, the components are analyzed and some primitive shape features of the human face are used to decide which region is a face and which is not.



**Fig. 1.** An overview of the system

## 3 Skin Color Model

Experience proved that skin is an effective and robust cue for face detection. Color is highly invariant to geometric variations of the face and it allows fast processing. Different colorspace were studied by different researchers in order to find an optimal representation for the skin color distribution. A survey about pixel-based skin color detection can be found in [7]. Many researchers claim that other colorspace, such as YCbCr, [5], performs better than the RGB colorspace. However the authors of [8] argue that the separability between skin and non-skin classes is highest in the RGB colorspace and that dropping the illumination

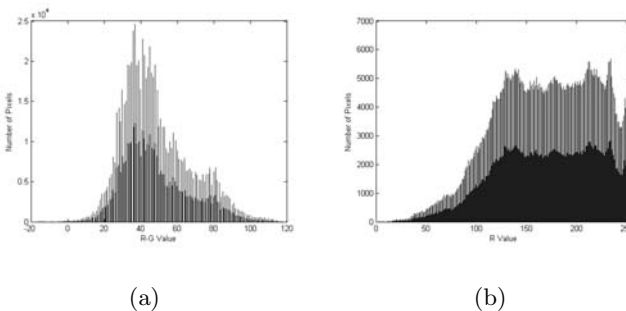
component worsens this separability. It is interesting to note here that an illumination compensation model is available in [5]. Many papers on skin detection do not properly justify the colorspace of their choice probably because it is possible to obtain good results, for a limited dataset, in any colorspace [7]. The performance of a skin detection method depends on a combination of the colorspace and the classifier, and we believe that it is possible to find an acceptable detector for any colorspace.

In this section, we present a classifier for skin color based in the RGB colorspace. The proposed algorithm is shown below:

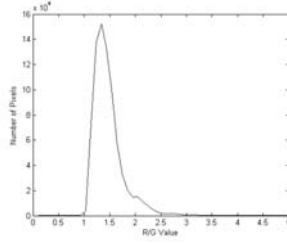
$$\begin{aligned}
 &(R, G, B) \text{ is classified as skin if:} \\
 &20 < R - G < 90 \\
 &R > 75 \\
 &R/G < 2.5
 \end{aligned}$$

The advantages of this method are its simplicity and computational efficiency since no transformation is needed to go to another colorspace. It is based on an idea presented in [9], where it was noticed that when subtracting the red channel from the green channel, the skin pixels tend to have distinctive values-greater than the non-skin pixels. In our experiments, around 800,000 skin pixels were taken from 64 different images of people with different ethnicity and under various light conditions. The distribution of R-G is shown in Fig. 2.(a), while that of R is shown in Fig. 2.(b), and that of R/G is shown in Fig. 3. Our experiments revealed that 94.6% of the skin pixels have their R-G values between 20 and 90, see Fig. 2.(a), which supports the observation in [9]. Also, we have noted that 96.9% of the skin pixels have their R value greater than 75, see Fig. 2.(b), and 98.7% of them have their R/G values less than 2.5, see Fig. 3. We found out that adding a constraint on R and another on R/G improves segmentation.

It should be noted that this algorithm was able to perform with lower noise than the one proposed by Kovac et al, [10], even though it has less constraints. An example of this segmentation is shown in Fig. 4.



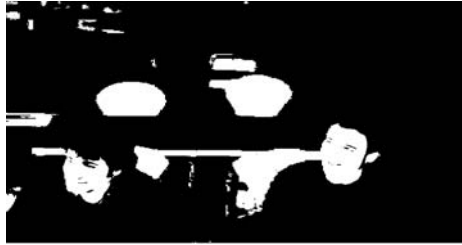
**Fig. 2.** (a) The distribution of R-G values in skin pixels (b) The distribution of R values in skin pixels



**Fig. 3.** The distribution of R/G values in skin pixels



(a)



(b)

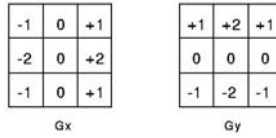
**Fig. 4.** (a) Original image (b) Segmented image

## 4 Face Localization

In this section, we will outline the steps we used in order to achieve the detection. Edge detection was used to separate the face from any non-face object. Later, the connected regions are grouped together and analyzed to determine which connected component is a face and which is not.

### 4.1 Edge Detection and Connected Components

As indicated earlier, edge detection is necessary to achieve a robust detection. A sobel edge detector, whose convolution masks are shown in Fig. 5, was used. The



**Fig. 5.** Sobel convolution masks

first mask estimates the gradient in the x-direction while the second estimates the gradient in the y-direction.

The gradient magnitude is calculated by:  $|G| = \sqrt{(G_x)^2 + (G_y)^2}$ .

An approximate magnitude which is faster to compute is given by:

$$|G| = |G_x| + |G_y|.$$

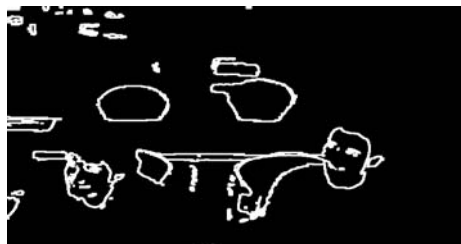
The angle which gives rise to the spatial gradient is computed through:

$$\Theta = \arctan(G_y/G_x).$$

After executing the edge detection, a dilation operation is applied to further separate the edges. The results of both processes are shown in Fig. 6.



(a)



(b)

**Fig. 6.** (a) Image with edges detected (b) Image after applying dilation

The resulting image, combining skin color segmentation with edge detection, is inverted (to assign the “1” value to the blobs rather than the edges) and searched for connected components according to the adjacent 8-neighbor pixels.



## 4.2 Face vs. Not-Face

Each of the connected components is then analyzed to judge whether it is a face or not. Simple shape features are used in the classification process. Each of these features, or cues, can be considered as a weak classifier. The cascade of these weak classifier forms a strong classifier, as demonstrated later in the experimental results. These features are:

1. Area: Normally, connected components with small areas correspond to noise generated by segmentation. Therefore, these regions are eliminated.
2. Bounding Box Proportions: The anatomy of the face suggests that the ratio of the bounding box height to its width is around 1.4 on average; however, this ratio varies slightly from one person to another. In our system, any region whose height is more than 1.8 times its width is removed.
3. Holes: From the geometry of the face, it is expected to see at least one hole in the region that corresponds to a face. Thus, any region with no holes is deleted.
4. Orientation: There is a limit on how much we can “pan and tilt” our heads. Therefore, it is logical to expect that the orientation of any face blob, with respect to the x-axis, is- in absolute value -between  $15^\circ$  and  $90^\circ$  approximately. Any blob whose orientation is outside this range is dropped.
5. Centroid: The face is evenly distributed in the region where it is located. Therefore, the centroid of a face region should be found in a small window centered in the middle of the bounding box. The dimensions of this window were found to be around 15% of the dimensions of the bounding box. Any region whose centroid is outside this window corresponds to a blob that is not evenly distributed and therefore it is not a face.
6. Extent: The extent of a blob is the area of this blob divided by the area of the bounding box (both in pixels). Given the elliptical form of the face and its distribution, our experiments revealed that the extent for a face is between 0.3 and 0.8. Thus, any region whose extent is not in this range is eliminated.

## 5 Experimental Results

Many of the face databases commonly used by researchers include only gray-scale images, such as FERET face recognition database and the CMU face detection database [11]. Actually, the field of face detection lacks a database where the same agents are photographed under varying poses. Therefore, we have tested our method on images generated by a sequence we recorded. The scenario of this sequence includes one person entering a cafeteria, and then he is followed by two of his friends. The three of them sit together and chat for a while before they leave. The advantage of this sequence is that it provides us with faces having variations in pose, size, position, and expression.

We have collected 211 sample images, containing 266 faces, from our recorded sequence, and we have tested our method on. Out of the 266 faces, 238 were

**Table 1.** Results Summary

Number of Faces	Positive Detections	Detection Rate	False Positives	Precision
266	238	89.45%	22	91.5%

correctly detected with 22 false positives. The detection rate is 89.5% and the precision is 91.5% proving that our method is robust and efficient in detecting faces. Precision is defined as the ratio of detected faces to the sum of detected faces and false positives. The results are summarized in Table 1. Some of the tested images are shown in Fig. 7. Please note the variations in pose, scale, position, and expression. Obtaining such a high detection rate on images under such variations is considered a good result.

**Fig. 7.** Experimental Results

## 6 Conclusions

In this paper, a novel approach for face detection in color images is proposed. Skin color segmentation is applied to separate the skin areas from the non-skin. Edge detection with dilation is then implemented to separate face candidates from any background or foreground blob. Connected components are later analyzed

using primitive shape features to decide which blob is a face and which is not. The experimental results revealed the robustness and efficiency of this method under varying conditions. Future work includes expanding this method to detect the different facial features.

**Acknowledgement.** This work is supported by EC grants IST-027110 for the HERMES project and IST-045547 for the VIDI-video project, and by the Spanish MEC under projects TIN2006-14606 and DPI-2004-5414. Jordi Gonzàlez also acknowledges the support of a Juan de la Cierva Postdoctoral fellowship from the Spanish MEC. The authors thank Dr. Ramon Baldrich for providing valuable assistance to this paper.

## References

1. Yang, J., Kriegman, D.J., and Ahuja, N., "Detecting Faces in Images: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 1, pp. 34-58, 2002.
2. Viola, P., and Jones, M., "Robust Real-time Object Detection", *Proceedings of IEEE Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, July 2001.
3. Huang, C., Ai, H., Wu, B., and Lao, S., "Boosting Nested Cascade Detector for Multi-View Face Detection", *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, vol. 2, pp. 415-418, August 2004.
4. Nara, Y., Yang, J., and Suematsu, Y., "Face Detection Using the Shape of Face with Both Color and Edge", *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, vol. 1, pp. 147-152, December 2004.
5. Ravi Kumar, C.N., and Bindu, A., "An Efficient Skin Illumination Compensation Model for Efficient Face Detection", *Proceedings of IEEE 32nd Annual Conference on Industrial Electronics*, pp. 3444-3449, November 2006.
6. Hsu, R.L., Abdel-Mottaleb, M., and Jain, A.K., "Face Detection in Color Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 5, pp. 696-706, 2002.
7. Vezhnevets, V., Vassili, S., and Andreeva, A., "A Survey on Pixel-Based Skin Color Detection Techniques", *Proceedings of International Conference on Computer Graphics (GRAPHICON'03)*, Moscow, Russia, September 2003.
8. Shin, M.C., Chang, K., and Tsap, L.V., "Does Colorspace Transformation Make Any Difference on Skin Detection?", *Proceedings of 6th IEEE Workshop on Applications of Computer Vision (WACV'02)*, pp. 275-279, December 2002.
9. Al-Shehri, S.A., "A Simple and Novel Method for Skin Detection and Face Locating and Tracking", *Lecture Notes in Computer Science, Computer Human Interaction*, vol. 3101, pp. 1-8, 2004.
10. Kovac, J., Peer, P., and Solina, F., "Human skin color clustering for face detection", *Proceedings of IEEE Region 8 Computer as a Tool (EUROCON'02)*, vol. 2, pp. 144-148, September 2003.
11. Zhang, H., and Zhao, D., "Spatial Histogram Features for Face Detection in Color Images", *Lecture Notes in Computer Science, Advances in Multimedia Information Processing (PCM'04)*, vol. 3331, pp. 377-384, 2004.

---

# Detection of Near-Duplicated Image Regions

Babak Mahdian<sup>1</sup> and Stanislav Saic<sup>2</sup>

<sup>1</sup> Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering, Břehová 7, 115 19 Prague 1, Czech Republic  
mahdian@utia.cas.cz

<sup>2</sup> Academy of Sciences of the Czech Republic, Institute of Information Theory and Automation, Pod Vodárenskou věží 4, 18208 Prague 8, Czech Republic  
ssaic@utia.cas.cz

**Summary.** Modern, easy to use image processing software enables forgeries that are undetectable by the naked eye. In this work we propose a method to automatically detect and localize near-duplicated regions in digital images. The presence of near-duplicated regions in an image may signify a common type of forgery called copy—move forgery. The method is based on blur moment invariants, which allows successful detection of copy—move forgery, even when blur degradation, additional noise, or arbitrary contrast changes are present in the duplicated regions. These modifications are commonly used techniques to conceal traces of copy—move forgery. Our method works equally well for lossy format such as JPEG.

## 1 Introduction

In this work we focus on detecting a common type of digital image forgery, called copy—move forgery. In copy—move forgery, a part of the image is copied and pasted into another part of the same image, with the intention to hide an object or a region of the image. Figure 1 shows an example. We can determine whether an image contains this type of forgery by detection of duplicated regions. Duplicated regions may not always match exactly. For example, this could be caused by a lossy compression algorithm, such as JPEG, or by possible use of the retouch tool.

Existing copy—move forgery detection methods are mostly based on matching of overlapping image blocks. For example, Fridrich et al. [4] has proposed a method which is based on matching the quantized lexicographically sorted discrete cosine transform (DCT) coefficients of overlapping image blocks. The lexicographically sorting of DCT coefficients is carried out mainly to reduce the computational complexity of the matching step. Another method has been proposed by Popescu and Farid [5] and is similar to [4]. This method differs from [4] mainly in the representation of overlapping image blocks. Here, the principal component transform (PCT) has been employed in place of DCT. The representation of blocks by this method has better discriminating features.

As pointed out in [4], ideal regions for using copy—move forgery are textured areas with irregular patterns, such as grass. Because the copied areas will likely blend with the background it will be very difficult for the human eye to detect



**Fig. 1.** An example of a copy—move forgery. The original (left) and forged version (right).

any suspicious artifacts. Another fact which complicates the detection of this type of tampering is that the copied regions come from the same image. They therefore have similar properties, such as the noise component or color palette. It makes the use of statistical measures to find irregularities in different parts of the image impossible.

## 2 Detection of Near-Duplicated Regions

To detect the copy—move forgery we focus our aim on detection of near-duplicated regions in the image. Existing copy—move forgery detection methods have limited abilities. In most cases of forgery investigated, they were able to detect duplicated regions in the tampered image despite of the presence of an acceptable amount of noise. This is mainly caused due to a quantization step or a similarity threshold. Additionally, it allows for analysis images compressed with a lossy algorithm, such as JPEG. However, a skilled falsifier will be able to produce a work undetectable by these methods.

This can be, for instance, achieved easily by blurring. An experienced falsifier can use a simple 2D convolution of the duplicated region with a blur filter mask to make detection of forgery even more difficult. Thus, to improve the detection abilities of the current available approaches, we can describe analyzed regions by some features invariant to the presence of unknown blur. From a mathematical point of view, we are looking for a functional  $B$ , which is invariant with respect to blur degradation. In other words,  $B$  satisfies the condition  $B(f) = B(D(f))$ , where operator  $D$  denotes the blur degradation. Furthermore, due to the fact that the falsifier can also use additive noise to make detection more difficult, these invariants should also work well with the presence of additive noise.

The aforementioned requirements are satisfied by blur moment invariants. They have been previously addressed by Flusser and Suk [1, 2] and have found successful applications in many areas of image processing such as: in face recognition on out-of-focused photographs, template-to-scene matching of satellite images, in focus/defocus quantitative measurement, etc. An advantage of moment invariants is that they are computed by a summation over the whole image, so they are not significantly affected by additive zero-mean noise.

We will define the problem of copy—move forgery detection in the following way. Given an image  $I(x, y)$  containing an arbitrary number of near-duplicated regions of unknown location and shape, our task is to determine the presence of such regions in the image and to localize them. The aim of this investigation is create a method that can detect duplicated regions, even when some contain degradations caused by convolution with a shift-invariant symmetric energy-preserving point spread function (PSF) and additive random noise. Formally: let  $f(x, y)$  be a function describing the original region and  $g(x, y)$  the acquired region created by the falsifier via convolution of  $f(x, y)$  with the PSF, then  $g(x, y) = (f * h)(x, y) + n(x, y)$ , where  $h(x, y)$  is a shift invariant PSF,  $n(x, y)$  an additive random noise and  $*$  denotes a 2D convolution. We would like to find all  $g(x, y)$  created from  $f(x, y)$  and  $h(x, y)$  via the above equation. Due

to the fact that moment invariants are utilized as features, we will assume the following restrictions. Both  $f(x, y) \in L_1$  and  $g(x, y) \in L_1$  are real functions and have a bounded support and nonzero integral. Moreover, the PSF is assumed to be axial symmetric and energy-preserving, i.e.:  $h(x, y) = h(-x, y) = h(y, x)$  and  $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y) dx dy = 1$ . These assumptions do not cause a significant limitation. Most imaging systems that we are interested in perform some type of symmetry. By supposing other types of symmetries, like central, four-fold or circular symmetry, we can also construct blur invariants based on moments. However, generally, the higher degree of symmetry of the PSF is assumed, the more invariants can be obtained [1].

The proposed copy—move forgery detection method is based on a few main steps: tiling the image with overlapping blocks, blur moment invariants representation of the overlapping blocks, principal component transformation, k—tree representation, blocks similarity analysis and near-duplicated regions map creation. Each step is explained separately in the following sections.

### 2.1 Overlapping Blocks

This method begins with the image being tiled by blocks of  $R \times R$  pixels. Blocks are assumed to be smaller than the size of the duplicated regions, which have to be detected. Blocks are horizontally slid by one pixel rightwards starting with the upper left corner and ending with the bottom right corner. The total number of overlapping blocks for an image of  $M \times N$  pixels is  $(M - R + 1) \times (N - R + 1)$ . For instance, an image with the size of  $640 \times 480$  with blocks of size  $20 \times 20$  yields 286281 overlapping blocks.

### 2.2 Blur Invariants Representation

Each block is represented by blur invariants, which are functions of central moments. The two-dimensional  $(p + q)$ th order moment  $m_{pq}$  of image function  $f(x, y)$  is defined by the integral:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

The two-dimensional  $(p + q)$ th order central moment  $\mu_{pq}$  of  $f(x, y)$  is defined as

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_t)^p (y - y_t)^q f(x, y) dx dy$$

where the coordinates  $(x_t, y_t)$  given by the relations  $x_t = m_{10}/m_{00}$ ,  $y_t = m_{01}/m_{00}$  denote the centroid or the center of gravity of  $f(x, y)$ . By supposing that  $g(x, y) = (f * h)(x, y)$ , we can simply derive that central moments of  $g(x, y)$  are defined as  $\mu_{pq} = \sum_{k=0}^p \sum_{j=0}^q \binom{p}{k} \binom{q}{j} \mu_{kj}^{(f)} \mu_{p-k, q-j}^{(h)}$ .

We are looking for features invariant to blur. Feature  $B$  is called blur invariant if  $B^f = B^{f * h} = B^g$ .

As mentioned, we consider only symmetric  $h(x, y)$ . By applying the algorithm as derived and described in [1, 3], we can construct blur invariants based on central moments of any order by using the following recursive relation:

$$B(p, q) = \mu_{pq} - \alpha \cdot \mu_{pq} - \frac{1}{\mu_{00}} \sum_{n=0}^K \sum_{i=m_1}^{m_2} \binom{p}{t-2i} \binom{q}{2i} \\ \cdot B(p-t+2i, q-2i) \mu_{t-2i, 2i}$$

where

$$K = [(p+q-4)/2], \quad t = 2(K-n+1), \quad m_1 = \max(0, [(t-p+1)/2]), \\ m_2 = \min(t/2, [q/2]), \quad \alpha = 1 \Leftrightarrow p \wedge q \text{ are even}, \quad \alpha = 0 \Leftrightarrow p \vee q \text{ are odd.}$$

The proposed algorithm uses 24 blur invariants up to the seventh order to create the feature vector  $B = \{B_1, B_2, B_3, \dots, B_{23}, B_{24}\}$  of each block. Some examples of utilized invariants in their explicit forms are listed below:

$$B_1 = \mu_{11}, B_2 = \mu_{12}, B_3 = \mu_{21}, B_4 = \mu_{03}, B_5 = \mu_{30}, \\ B_6 = \mu_{13} - \frac{3\mu_{02}\mu_{11}}{\mu_{00}}, B_7 = \mu_{31} - \frac{3\mu_{20}\mu_{11}}{\mu_{00}}, \\ B_8 = \mu_{32} - \frac{3\mu_{12}\mu_{20} + \mu_{30}\mu_{02}}{\mu_{00}}, B_9 = \mu_{23} - \frac{3\mu_{21}\mu_{02} + \mu_{03}\mu_{20}}{\mu_{00}}.$$

Because we will use an Euclidean metric space, the invariants should be normalized to have the same weight. To achieve this, the normalization described in [2, 3] is used  $B'_i = \frac{B_i}{(R/2)^r \mu_{00}}$ , where  $R$  is the block size and  $r$  the order of  $B_i$ . Please note that in this manner normalized invariants are also invariant to contrast changes, which improves the duplication detection abilities of the algorithm.

As is obvious, each block is represented by a feature vector of length 24 in the case of gray-scale images. For RGB images, moment invariants of each block in each channel are computed separately, resulting in feature vector  $B_{rgb} = \{B_{red}, B_{green}, B_{blue}\}$  of length 72.

### 2.3 Principal Component Transformation

In the case of an RGB image, the dimension of the feature vector is 72 (24 invariants per channel). Using the principal component transformation we reduce this dimension. Fraction of the ignored variance along the principal axes is set to 0.01. In PCT, the orthogonal basis set is given by the eigenvectors set of the covariance matrix of the original vectors. Thus, it can be easily computed on very large data sets. Note that PCT preserves the Euclidean distance among blocks.

### 2.4 k—d Tree Representation

In blocks similarity analysis (see next section) we will need to efficiently identify all blocks which are in a desired similarity relation with each analyzed block.



A simple exhaustive search computes the distance from the block to all others. This approach is very inefficient and its computational cost is  $O(N)$ . To improve the efficiency of finding neighboring blocks, some hierarchical structures have been proposed. The  $k$ - $d$  tree is a commonly used structure for searching for nearest neighbors.

The  $k$ - $d$  tree preprocesses data into a data structure allowing us to make efficient range queries. It is a binary tree that stores points of a  $k$ -dimensional space in the leaves. In each internal point, the tree divides the  $k$ -dimensional space into two parts with a  $(k-1)$ -dimensional hyperplane. If a  $k$ - $d$  tree consists of  $N$  records, it requires  $O(N \log_2 N)$  operations to be constructed and  $O(\log_2 N)$  to be searched. Because of these reasons, the proposed method transforms blocks representation to a  $k$ - $d$  tree for a more effective closest neighbors search.

## 2.5 Blocks Similarity Analysis

The main idea of this step is that a duplicated region consists of many neighboring duplicated blocks. If we find two similar blocks in the analyzed space and if their neighborhoods are also similar to each other, there is a high probability that they are duplicated and they must be labeled.

The similarity measure  $s$  employed here is defined by the formula:  $s(B_i, B_j) = 1/(1 + \rho(B_i, B_j))$ , where  $\rho$  is a distance measure in the Euclidean space  $\rho(B_i, B_j) = (\sum_{k=1}^{dim} (B_i[k] - B_j[k])^2)^{1/2}$ . For each analyzed block represented by the feature vector  $B$ , we look for all blocks with an equal or larger similarity relation. It must be an equal or larger similarity to the threshold  $T$ . The method finds all similar blocks for each one (similar to the nearest neighbors search) and analysis their neighborhood. This is done efficiently using the  $k$ - $d$  tree structure, which was created in the previous step.

If  $s(B_i, B_j) > T$ , where  $T$  is the minimum required similarity, we also analyze the neighborhood of  $B_i$  and  $B_j$ . Note that the threshold  $T$  plays a very important role. It expresses the degree of reliability with which blocks  $i$  and  $j$  correspond with each other. It is obvious that the choice of  $T$  directly affects the precision of results of the method. Due to the possibility of the presence of additive noise, a boundary effect, or JPEG compression, this threshold should not be set to 1.

After two blocks with the required similarity have been found, a verification step begins. In the verification step, similar blocks with different neighbors will be eliminated.

For analyzing the blocks neighborhood, we choose 16 neighboring blocks with a maximum distance of 4 pixels from the analyzed block (distance from their upper left corners). If 95% of these neighboring blocks satisfy the similarity condition, the analyzed block is labeled as duplicated. More formally, block 1 with coordinates  $(i, j)$  and block 2 with coordinates  $(k, l)$  are labeled as duplicated if  $s(block(i + x_r, j + y_r), block(k + x_r, l + y_r)) \geq T$ , where  $x_r \in \langle -4, -3, \dots, 3, 4 \rangle$  and  $y_r \in \langle -4, -3, \dots, 3, 4 \rangle$  and  $r = 1 \dots 16$ . This part of the algorithm also determines the minimum size of the copied area, which can be detected by the algorithm.

To have more precise results, the verification step additionally uses information about the image distances of analyzed blocks.

If  $s(\text{block}(i, j), \text{block}(k, l)) \geq T$ , but  $\sqrt{(i - k)^2 + (j - l)^2} \leq D$ , these blocks will not be further analyzed and will not be assigned as duplicated. Threshold  $D$  is a user-defined parameter determining the minimum image distance between duplicated regions.

The output of this section is matrix  $Q$  with the same size as the input image. Elements of this matrix are either zero or one. An element of this matrix is set to one if the block at this position is duplicated.

## 2.6 Near-Duplicated Regions Map Creation

The output of the method is a near-duplicated regions map showing the image regions, which are likely duplicated. It is created by the multiplication of each element of  $I(x, y)$  by its respective element in  $Q(x, y)$ . Matrix  $Q(x, y)$  is created in the previous section.

## 3 Experimental Results

An experimental version of the proposed method was implemented in Matlab. The output of the method is a duplication map, in which likely duplicated regions are shown. Parameters of the method were set to  $R = 24$  (block size),  $T = 0.98$  (similarity threshold),  $D = 24$  (blocks image distance threshold). These parameters can be changed (for instance,  $R = 20$ ,  $T = 0.97$ ), but it has a strong influence on obtained results. In the PCT step, the fraction of the ignored variance along the principal axes,  $\epsilon$ , was set to 0.01. Please note that the computational time of the method is highly dependent on these parameters (specially on  $T$  and  $\epsilon$ ). In the presented example, the tampering was realized by copying and pasting a region in the image with intent to conceal a person or object. Additionally, in order to make the detection of forgery more difficult and interesting, the second example contains further manipulations of the pasted region.

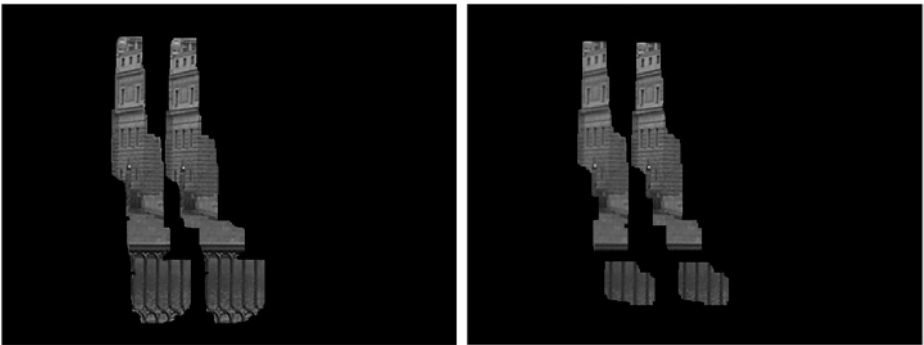


Fig. 2. Two duplication maps constructed by the proposed method

The first example is presented in the top image of Figure 2. It shows the output of the method applied to the tampered image shown in Figure 1(b). In this example, no further manipulations were carried out with the tampered regions. The tampered image was saved in TIFF format. The output shows that the proposed method correctly detected the near-duplicated regions.

The bottom image of Figure 2 shows the duplication map created by applying our method to Figure 1(b). In this example, the tampered region was blurred with a Gaussian blur filter with radius 0.3 pixels. The tampered image in this case was saved in JPEG format quality 80.

## 4 Discussion

Our results show that the use of blur moment invariants can improve the detection abilities of the copy—move forgery detection methods. By using blur moment invariants we are able to additionally detect duplicated regions with presence of acceptable blur and additive Gaussian noise. By normalizing moment invariants they also become invariant against contrast changes. The proposed method also works with lossy JPEG format images.

It can be interesting to mention briefly the stability of moment invariants under additive random noise. As aforementioned, moment invariants are computed by a summation over the whole image, so they are not significantly affected by additive zero-mean noise. For a more detailed discussion about this topic see [3].

Our method is based on truncated versions of the filtered blocks which are additionally corrupted by the neighboring pixels. Therefore we have to mention the stability of moment invariants with respect to boundary effect. If our blocks have  $R \times R$  pixels and the size of PSF support is  $H \times H$  pixels, the correct size of the resulting block must be  $(R + H - 1) \times (R - H - 1)$ . In our case, the value of  $H$  is unknown. If  $H \ll R$  the errors of invariant calculation caused by the boundary effect is negligible. If  $H$  is relatively large as in the case of heavy blur, the boundary effect will cause significant miscalculations of the invariant values. For an experiment on this topic, see [3].

A disadvantage of the proposed method is its computational time. The average run time of the implemented experimental version with parameters  $R = 24$  (block size) and  $T = 0.98$  (similarity threshold) for  $640 \times 480$  RGB images on a 2.1 GHz processor and 512 MB RAM is 30 min. The computational time is not the same for images with the same size. It is dependent on each images characteristics (the dimension of space created after the PCT) and especially on the similarity threshold parameter of the algorithm. It is also important to note that the implemented experimental version was not optimized and there exist possibilities to improve the computational time.

A way to considerably improve the computational time is to eliminate large uniform areas in a preprocessing step and apply the proposed method to the rest of the analyzed image. Then the output of the method could consist of a duplication map and a uniform areas map.

## References

1. J. Flusser and T. Suk. Degraded image analysis: An invariant approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):590–603, 1998
2. J. Flusser, T. Suk, and S. Saic. Image features invariant with respect to blur. *Pattern Recognition*, 28(11):1723–1732, 1995
3. J. Flusser, T. Suk, and S. Saic. Recognition of blurred images by the method of moments. *IEEE Transactions on Image Processing*, 5(3):533–538, 1996
4. J. Fridrich, D. Soukal, and J. Lukas. Detection of copy-move forgery in digital images. In *Proceedings of Digital Forensic Research Workshop, Cleveland, OH, USA, August 2003*, IEEE Computer Society
5. A. Popescu and H. Farid. Exposing digital forgeries by detecting duplicated image regions. Technical Report TR2004-515, Department of Computer Science, Dartmouth College, 2004

---

# Automatic Merging of 3D Attribute Meshes

Krzysztof Skabek and Agnieszka Tomaka

Instytut Informatyki Teoretycznej i Stosowanej PAN, Gliwice, Poland  
{kkskabek, ines}@iitis.gliwice.pl

**Summary.** In the paper we introduce the method of automatic merging of 3D attribute meshes. Attributes describe the certainty of particular vertices/faces in the mesh. We calculate values of attributes on the basis of the assumed observation strategy as well as the properties of the scanning system. We show the resultant meshes and compare the method to the existing solutions.

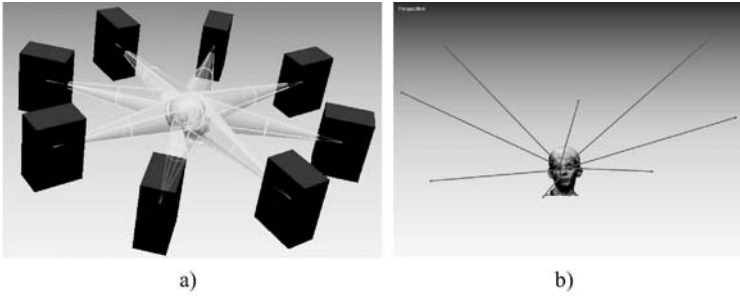
## 1 Introduction

The 3D laser scanner Minolta VI9i can precisely obtain data of the surfaces which are visible from its observation point. In order to scan the whole object a number of measurements have to be done from different observation points around the object. Our task was to integrate the observation data into one surface model. So far this process was performed semi-automatically by a qualified person whose task was to choose the most reliable points and faces in scans obtained from particular directions. This tiresome process was followed by registration and merging of scans and then visual verification of the resulting model surface. When the merged surface was folded, faulty or too many holes appeared, the whole process had to be repeated. Even if the results are acceptable the repeatability of the process cannot be guaranteed. Sometimes the process can be facilitated using a rotary stage. Then the information of the stage rotation is used to find the initial registration of input range images in the surface model. But the difficulties are much more serious when we measure unstable objects such as human heads. Such situation occurs in our research as we consider building 3D surface models of human heads to support orthodontic diagnosis [5].

The scanner accuracy and the scanning procedures for orthodontic diagnosis were discussed in [3]. The paper concluded that the best way to obtain full coverage of the human head is to choose 7 observation points plus 1 rear view placed evenly on the observation circle as shown in fig. 1a. Such number of scans yields some redundancy but assures that each important part of the head surface is scanned.

## 2 Automatic Surface Integration

The automatic registration and merging of range images of a human head into a 3D surface model can be decomposed into three steps: (1) processing the input

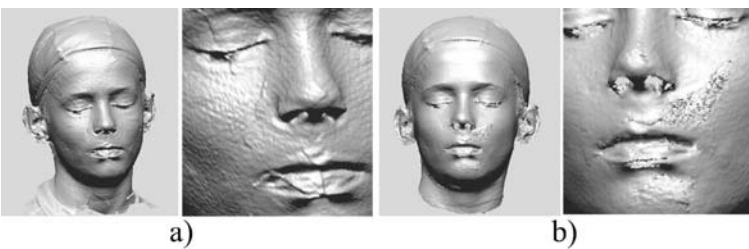


**Fig. 1.** a) positioning of the observation points, b) observation vectors for particular input range images

range images, (2) registration of input meshes and (3) merging input meshes into one surface model. In the paper [1] the stages of initial processing and registration were described and here we deal mainly with attribute data merging that corresponds to the last integration step.

In the automatic surface integration we need first to roughly position the input range images in the global coordinate system even before the initial registration. Such positioning is performed on the basis of assumptions concerning the measurement procedure. The observation vectors of the scanning device are defined and computed for particular observation points (see fig. 1b). These vectors are also used to compute weights of the surface mesh model.

The registration of input meshes uses points as matching units with the point-to-point distance metric. Basically the algorithm uses a framework of the pairwise ICP algorithm [2]. The aim is to minimize globally an objective function for all input meshes.



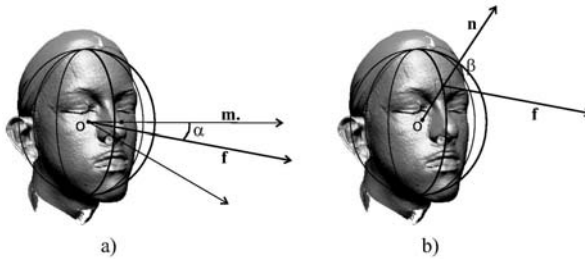
**Fig. 2.** Artifacts on the merged surfaces: a) surface merging, b) volume merging

Unfortunately the initial registration is not enough to do the proper merging using surface or volume methods because some artefacts (downcasts, granules, etc.) appear in places of the overlapping surfaces (see fig. 2). However, such artefacts are not the result of faulty behaviour of the merging methods, but they

depend on: noises in range images, the limited accuracy of the scanning device especially for strongly inclined faces, the angular distance from the scanner axis or simply movements of the patient during the scanning. In what follows, we deal with such problems (except the last one) by (1) eliminating the redundant structures, (2) selecting the most reliable parts of the input meshes to register and merge and, finally, subjection of the merge process to a reliability (weight) of vertices in input meshes.

### 3 Valuating Uncertainty of Input Meshes

At this point we can distinguish two ways of uncertainty assessment of input meshes: (1) discriminating faces having the greater angle from the vertical plane containing the observation line (see fig. 3a), (2) discriminating faces inclined to the observation axis stronger than a given angle (see fig. 3b). Of course both measures concern the weights of faces in the input mesh but weights can easily be computed for each vertex as an average of face weights adjacent to it. The first point is extended to the task of range image segmentation.



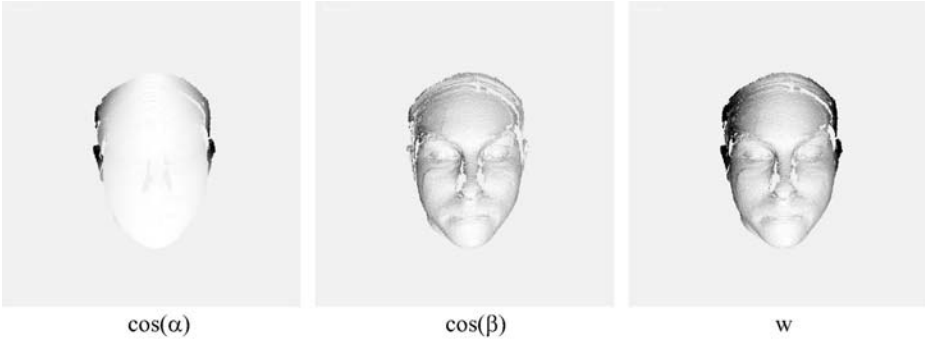
**Fig. 3.** Valuating uncertainty of input meshes: a) a face angular distance to the vertical observation plane, b) a face inclination to the observation line

In practice values of weights for each face in the input mesh are computed from the scalar product dependence and particularly:

$$\cos \alpha = \frac{\mathbf{m} \cdot \mathbf{f}}{|\mathbf{m}| \cdot |\mathbf{f}|} \quad \text{and} \quad \cos \beta = \frac{\mathbf{n} \cdot \mathbf{f}}{|\mathbf{n}| \cdot |\mathbf{f}|}.$$

And the resultant value of weight for faces is defined as:  $w = \cos \alpha \cdot \cos \beta$ ,  $w \in [0, 1]$ .

We constructed maps of weights for a given example of the input mesh (see fig. 4) to illustrate a distribution of weight values. The intensity of points in maps is proportional to weight values and their range is from 0 (black colour) to 1 (white colour). Note that the scalar product for  $\cos \alpha$  is in fact calculated for vectors  $\mathbf{n}'$  and  $\mathbf{f}'$  which are the planar projection of vectors  $\mathbf{n}$  and  $\mathbf{f}$  in the horizontal plane.



**Fig. 4.** maps of weights for an example of the input mesh



**Fig. 5.** Preliminary segmentation of input meshes

## 4 Segmentation of Input Meshes

Our approach makes it possible to divide the input meshes into strips regarding the reliability of surfaces (fig. 5). The reliability of surfaces is proportional to the angular distance to the vertical plane containing the scanning vector. Such strategy is connected to a feature of the scanning device that the best accuracy is obtained in the middle of the scanning area [1]. The segments of the input mesh were selected in a discrete manner so that we distinguish strips with reliability up to the value 0.3. This makes the most reliable strips that are taken to the final global registration and the attribute merging. Each strip is taken to the fine registration with a piece of the neighbouring structure wide enough to provide the overlapping area for registration [2].

## 5 Merging Attribute Meshes

The merge process incorporates three stages: (1) merging of faces, (2) merging of vertices and (3) merging of texture. The first two stages are closely connected and the third is rather a separate process. As an input we take weighted mesh structures for each observation point of the scanner and sequentially integrate them into a model mesh.

First we find the corresponding faces in the model mesh for each face of the input mesh. The centers of mesh faces are calculated beforehand and used



to create a KD-tree structure [4] for model mesh faces. The KD-tree is quickly searched for face-to-face correspondences and when the correspondence is formed the proper vertex-to-vertex matching is chosen for the corresponding faces. In fact there are only three combinations of vertex matchings as we assume that normal vectors of the corresponding faces have the same senses. If the distance between faces is less than a given threshold  $d_1$  and the average distance of the corresponding vertices is also less than a threshold  $d_2$  then we combine these faces and vertices into one face; otherwise, we add a new face to the model structure. New vertices are added also provided that the corresponding vertices do not exist. In our case it was assumed that  $d_2 = 3d_1$ .

The integration of many vertices from the input structures into the model mesh causes modification of their position. A new position is computed considering the weights of component vertices.

The colour attribute is calculated for each vertex of the model mesh as a weighted average of component input vertices proportionally to their weight attributes.

We use other parameters beside the distance thresholds ( $d_1, d_2$ ) to increase the accuracy of the merging process. One of them is *weight cut-off* parameter  $k \in [0, 1]$ . It makes it possible to reject vertices of weight  $w < k$ . In this case we get rid of structures that were acquired imprecisely. There is another parameter  $p$  which allows us to remove not proportional faces out of the model mesh. As not proportional we mean such faces that have the shortest edge  $p$  times smaller than the longest one. In our approach we assumed the following default values of the parameters:  $d_1 = 2(\text{mm})$ ,  $k = 0.2$ ,  $p = 4$ . The values were chosen experimentally for our testing data to obtain the best accuracy.

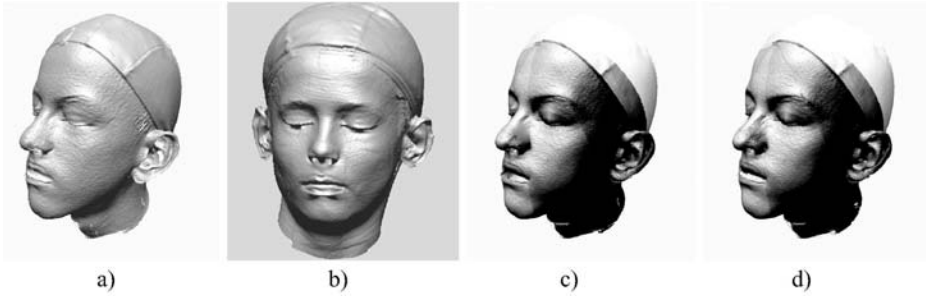
## 6 Results

We implemented the method for merging attribute meshes and tested it on several cases of 3D face measurement. The complete mesh models with texture are presented in fig. 6.

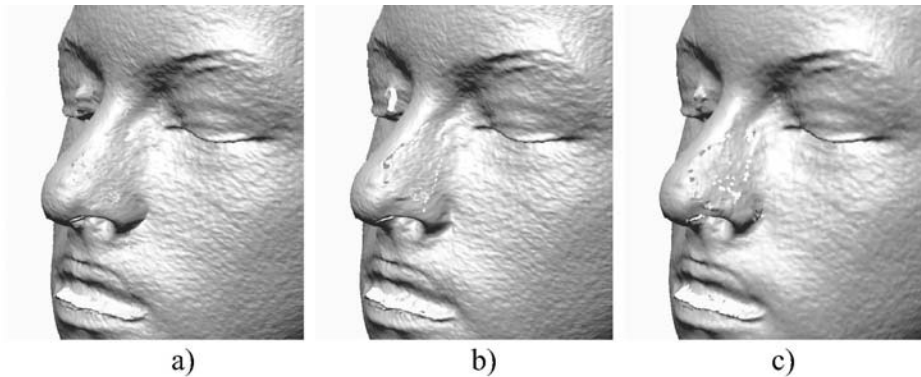
As the merging method has been implemented in the RapidForm programming environment we tested various methods to enhance the merging results. One of them is the function *fill holes*. The function makes it possible to complete the model structure even if it was not completely obtained from any observation point or was obtained with significant noise.

RapidForm provides two standard functions to merge meshes: *surface merge* and *volume merge*. We compared them to our attribute merge method (see fig. 7).

The main advantages of the described integration method are: (1) automatic integration of range images facilitates the tiresome work of technician, (2) repeatability - for the same range images and the same parameters the result is the same. The integration algorithm depends mainly on the quality of registration and a stable position of patients is of the great importance. Some downcasts can appear in case of movements of the patient during the examination.



**Fig. 6.** a), b) examples of merged faces, c) coloured and filled face merged using attribute method, d) coloured and filled face merged using surface method



**Fig. 7.** Comparison of merging methods: a) attribute merging, b) surface merging, c) volumetric merging

## Acknowledgement

This work was supported by the Ministry of Science and Higher Education of Polish Government, the research project no 3 T11C 028 27.

## References

1. Tomaka A, Skabek K (2006) Automatic Registration and Merging of 3D Surface Scans of Human Head. In: Piętka E, Łęski J, Franiel S (eds) *Medical Informatics & Technology*: 75–81
2. Nishino K, Ikeichi K (2002) Robust Simultaneous Registration of Multiple Range Images. *The 5th Asian Conference on Computer Vision*, Melbourne, Australia

3. Tomaka A, Luchowski L, Skabek K, Tarnawski M (2005) 3D Head Surface Scanning Techniques for Orthodontics. *Journal of Medical Informatics & Technologies*, vol. 9: 123–130
4. Samet H (2006) *Foundations of Multidimensional and Metric Data Structures*. The Morgan Kaufmann Series in Computer Graphics
5. Tomaka A, Liśniewska-Machorowska B (2005) The Application of the 3D Surface Scanning in the Facial Features Analysis, *Journal of Medical Informatics & Technologies*, vol. 9: 233–240

**Features, Learning and Classifiers**

---

# Learning Machines Information Distribution System with Example Applications

Norbert Jankowski and Krzysztof Grąbczewski

Department of Informatics, Nicolaus Copernicus University, Toruń, Poland  
{norbert,kgrabcze}@is.umk.pl

**Summary.** When problem solving reduces to examination of a single or a few learning methods no sophisticated mechanisms of information exchange are necessary, but when we use meta-learning for extensive search through a huge space of hybrid models, the information exchange between subsequent models is crucial.

The information exchange between models must be universal, very flexible and as simple to define as possible. The design of an efficient system must include abstract methodology for transmission of amorphous information between different kinds of methods and optimizing different types of functions. It is highly important for meta-learning where different types of information must be collected and used by meta-processes at high level of abstraction. This article presents a universal information exchange system eligible for huge data mining tasks which was implemented in our meta-learning environment Intemi.

## 1 Introduction

Meta-learning [1, 2] techniques will more and more often supply successful models, which would be very difficult to find by human, because of their unusual structures. Some of our research have already born the fruits of very high accuracies of our classifiers solving tasks of the Feature Selection Challenge<sup>1</sup> [3] of NIPS 2003 and the Handwritten Digit Recognition Competition<sup>2</sup> organized with The Eighth International Conference on Artificial Intelligence and Soft Computing in 2006. The models we found were usually complex model structures consisting of some data transformations like standardization, feature selection, features construction based on principal components analysis and some committees of classifiers. We have also examined some aspects of member model competence in classification committees [4].

All such meta-learning machines (algorithms) require large amount of calculations (e.g. to validate the methods) before they point to most attractive solutions. Many candidates must be examined, numerous combinations validated often with different optimization criteria. To make it all possible we need a general

---

<sup>1</sup> <http://www.clopinet.com/isabelle/Projects/NIPS2003/>

<sup>2</sup> <http://www.icaisc.pcz.czest.pl/competition.htm>

data mining system, which efficiently manipulates such complex models. Such system must provide:

- uniform way of machines and model manipulation—the possibilities of adding, configuring, training and testing machines, exploiting and removing models in a standard way, implemented as a set of project management routines in such a way that does not burden the authors of particular machines with the administration efforts,
- **uniform access to results of learning and tests, so that meta-learning methods do not need knowledge about the specificity of particular models,**
- **uniform query system for gathering information from submachines, facilitating versatile and efficient functionality of analysis of gathered results.**

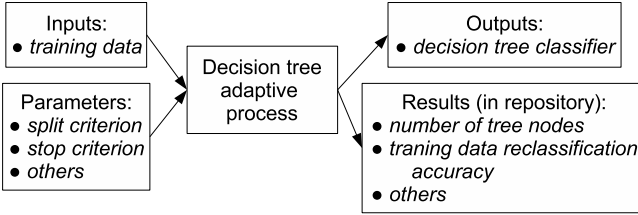
The second and third points define the main theme of this paper. The mechanisms must be uniform but not too restrictive, i.e. general enough to fit any kind of adaptive processes (also the results of machines which will be constructed in future). The abstraction of management routines facilitates communication between different machines and models within the project on appropriate, different levels of abstraction. Dependently on particular needs, general or detailed questions may be asked in a common language without the necessity to know the details of the methods being used. This provides an excellent source of knowledge (meta-knowledge) not only for basic analysis of datasets, but also for advanced meta-learning.

Section 2 sketches some ideas of the system with special emphasis on the topics of this article (more information is presented in [5]). Next, in section 3, we describe the abstraction of learning results representation. In section 4, the general methodology of exploring the results is presented and illustrated by examples in the last section 5.

## 2 Fundamental Ideas of Our Data Mining System

Our system is a general data mining tool eligible for any computational intelligence applications. There are many data mining systems available on the market, but we don't know any, providing so rich general functionality of the kernel and being so suitable for advanced meta-learning algorithms management. Abstraction of our system is based on generalized definitions of *method* (*machine*) and *model*, an abstract view of *inputs* and *outputs*, *parameters* and *results* and provides general tools for machine and model management, independent of the kind of the algorithm.

In computational intelligence, the term *method* (or *learning machine*) is used to describe an *adaptive algorithm*. In our approach this term encompasses a broader (than usual) range of algorithms, because from the point of view of a general data analysis framework there is no reason to differentiate between the algorithms for classification, approximation or clustering and those for loading



**Fig. 1.** Decision tree learning machine structure

data, visualizing some aspects of data, testing classifiers etc. We define a *model* as a result of application of a *machine* with some particular *parameters* to objects given as *inputs*. A model is an information carrier—this information may be passed to other models by means of *outputs* and may be put into a special *results repository*.

The aspects of accessing inputs, output exhibition and management, access to submachines and their configurations, results repository navigation etc. have been implemented in *MachineBase* class. The class is common to all possible machines and allows the implementers to work only on the crucial code for particular machines. Similar idea lies behind the configuration of the machine. Hence, we have created a general *ConfigBase* class implementing the common functionality related to machine configuration (inputs and outputs definition, parameters, submachines configurations) and available to learning machines developers.

Machines within a project compose two hierarchies. One is defined by the *input-output relation*. The other is defined by the *parent-child (or submachine/submodel) relation*.

The distinction between outputs and results is technical and concerns the way they can be used by external methods. Both are the effects of the adaptive process, but outputs are to be bound with other machines inputs, and the results are deposited in a special repository, which makes them available even after the model itself is released (for example when a vast amount of model structures is tested, and together they would occupy too much memory).

An example of the scenario with inputs, parameters, outputs and results is shown in figure 1. It depicts a decision tree adaptive process with single input of training data and some parameters. The model exhibits classification routine as an output (for other machines use) and deposits some numbers in the results repository.

### 3 Results Repository

To provide uniform results management, we created external (to the model) results repository containing items in the form of *label-value* pairs. The string *label* lets queries recognize the values, which can be *objects* containing information of any type. The object may be a number, string, collection of other values, etc.

For an example, consider a classification test. It should have two inputs: the classifier and the data to classify (including proper class labels, to be compared to the predicted ones), and at least two outputs: one exhibiting the labels calculated for the elements of the input dataset and one for the confusion matrix. The natural destination for the calculated accuracy is the results repository. To add the value of *accuracy* variable, labeled as "*Accuracy*", we need to call:

```
machineBase.AddToResultsRepository("Accuracy", accuracy);
```

where the *machineBase* is the object of *MachineBase* class corresponding to our test model. Note that each model has its own results repository node which is accessible for models of higher levels (parent models). Exactly in the same way any other model can add anything to the results repository, for example SVM can provide the value of its margin, an ensemble can inform about its internals etc.

Another useful possibility (especially for complex models) is that the addition of label-value items can be done from outside of the model. For instance, the parent model can add some information about its submachines, which depends on the context in which the child machine occurs, and which the child can not be aware of.

An example of such a parent is cross-validation (CV), which can label its submachines with information on *repetition* number and the CV *fold*:

```
submachineCaps.AddToResultsRepository("Repetition", repetition);  
submachineCaps.AddToResultsRepository("Fold", fold);
```

This labeling is performed in the context of model *capsule*, in which each model is placed because of some efficiency requirements, which are outside of the scope of this article. It may also be seen as labeling the connections between parent and children.

### *Commentators*

To extend the functionality of results repository, we came up with the idea of model *commentators*. It facilitates extending the information in results repository about particular models by external entities other than parent machines. The necessity of such solution comes from the fact that the author of the machine can not foresee all the needs of future users of the models and can not add all interesting information to the results repository. On the other hand it would not be advantageous to add much information to the repository, because of the danger of high memory consumption, which results repository is designed to minimize.

Commentators have access to machine's inputs and outputs, configuration and any *public* parts. In addition commentators may also calculate new values. Commentator may put to the repository a knowledge extracted from any part of the model, its neighborhood, or even from its submachines.

Commentator can be assigned to models by means of the *ConfigBase* class in a simple way:

```
classifierTestConfig.DeclareCommentator(new CorrectnessCommentator());
```



An example of useful commentator, defined for classification test, may be helpful in different statistical tests like McNemar's. To perform such tests, the information about the correctness of classification of all the instances of tested data is necessary (a vector of boolean values telling whether subsequent classifier decisions were right or wrong).

## 4 Query System

The aim of the methodology of results repository and commentators is to ensure that all the models in the project are properly described and ready for further analysis. Gathering adequate results into appropriate collections is the task of the *query system*.

The features of a functional query systems include:

- efficient data acquisition from the hierarchy of models,
- efficient grouping and filtering of the collected items,
- a possibility to determine pairs of corresponding results (for paired t-test etc.),
- a possibility of performing different transformations of the result collections,
- rich set of navigation commands within the results visualization application, including easy model identification for a result from a collection, navigation from collections to models and back, easy data grouping and filtering, etc.

The main idea of the query system is that the results repository, which is distributed throughout the project, can be searched according to a query, resulting in a collection called *series*, which then can be transformed in a wide spectrum of ways (by special components, which can be added to the system at any time to extend the functionality) providing new results which can be further analyzed and visualized.

All the ideas of *series*, their *transformations* and *queries* are designed as abstract tools, adequate for all types of models, so that each new component of the system (a classifier, test etc.) can be analyzed in the same way as the others, without the necessity of writing any code implementing the analysis functions.

### *Series*

The collection of results obtained from results repository as a result of a query is called *series*. In the system, it is implemented as a general class *Series*, which can collect objects of any type. Typically it consists of a number of information items, each of which, contains a number of values. For example, each item of the series may describe a single model of the project with the value of its classification accuracy, the number of CV fold in which the model was created etc. Thus each item is a collection of label-value pairs in the same way as in the case of results repository. Such representation facilitates two main functions of the series: grouping and filtering.

*Series transformations*

The series resulting from queries may not correspond right away to what we need. Thus, we have introduced the concept of *series transformations*. In general the aim of transformations is to convert a number of input series into a single output series. Some of the most useful transformations accessible in our system are:

- calculating properties (correlations, means, standard deviations, medians etc.) or statistical tests (t-test, McNemar test, Wilcoxon test, etc.),
- a concatenation of series into one series (e.g. for grouping together the results of two classifiers into a single collection),
- combining two (or more) series of equal length into a single series of items containing the union of label–value pairs from all the items at the same position in the input series,
- calculating different expressions on series.

*Queries*

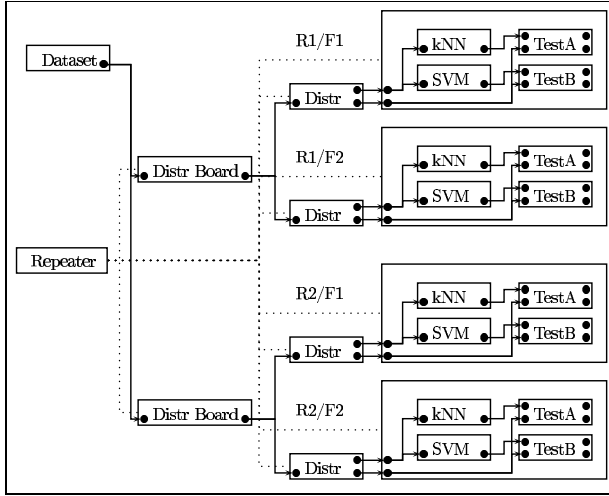
To obtain a series of results collected from results repository, we need to run a *query*. A query is defined by:

- the *root node*, i.e. the node of the project (in fact a model capsule), which will be treated as the root of the branch of the parent–child tree containing the candidate models for the query,
- the collection of *machine configurations* defining which models of the branch will actually be queried (the results are collected from models generated by machines using configurations from the collection),
- the *labels* to collect, which correspond to label–value pairs in results repository.

The result of running a query is a series: the collection of items corresponding to the models occurring in the branch rooted at the *root node* and being the results of running machines configured with one of the settings in *machine configurations* (see next section for illustrative examples). Each of the items is a collection of label–value pairs extracted for the collection of *labels*. For greater usefulness, the labels in the third parameter of the query, are searched not only within the part of results repository corresponding to the queried model, but also in the description of parent models (we will see the advantage of such a solution in the following examples).

## 5 Example Applications of Queries and Series Transformations

Consider the example of model structure presented in figure 2. It is a sketch of the hierarchy of models obtained with a repeater machine running twice 2-fold CV of two classifiers (in parallel) kNN and SVM. Each labeled box represents a single model. There are also four groups of classifiers and their tests enclosed by unnamed boxes—they are the scenarios run in each fold of the CV. The bullets in the left part of the boxes represent model inputs, and those at the right



**Fig. 2.** A repeater machine performed twice a 2-fold cross-validation of two classifiers

side—the outputs. The repeater model contains a sequence of runs resulting from the configuration of the distribution board. The two “Distr Board” boxes correspond to the CV distribution board, which splits its input dataset into 2 parts, preparing it for the CV. The “Distr” boxes are distributors—they use the distribution board output to exhibit proper training and test datasets as their outputs. The repeater created a scenario defined at the configuration stage for each of the distributors. The scenario assumed creation of kNN and SVM models with inputs bound to proper CV training data and one classification test for each of the two classifiers. The test machines’ inputs are bound to corresponding classifiers and CV test data respectively.

After the structure of models is created and the adaptive processes finished, different queries may explore the results repository. The most desirable query in such case is certainly the query for the collection of CV test classification results of each of the classifiers. To achieve this we may define the query(-ies) in the following way:

```
Query q = new Query();
q.Root = repeaterCapsule;
q.AddMachineConfig(testA);
q.MainLabel = "Accuracy";
knnSeries = q.Series;
```

```
Query q = new Query();
q.Root = repeaterCapsule;
q.AddMachineConfig(testB);
q.MainLabel = "Accuracy";
svmSeries = q.Series;
```

the *root node* is the repeater node, the *machine configurations* include *testA* and *testB* configurations respectively for kNN and SVM (this ensures that the results will be collected only from the appropriate test models), and the main label is

set to "Accuracy". In some cases like in the below example of McNemar test, we may prefer to set the *MainLabel* to "Correctness" in place of "Accuracy":

```
q.MainLabel = "Correctness";
```

Such queries return series (*knnSeries*, *svmSeries*) of four accuracies calculated by tests of KNN and SVM models respectively. Both the *Repetition* and the *CV-fold* labels are assigned to the connection leading to the box with classifiers (R1 or R2 and F1 or F2), so appropriate information must be included, but it is too technical to describe it here. As a result we obtain a series of four items consisting of the values of accuracy, repetition number and CV fold number.

#### *McNemar test of two classification models*

To test statistical difference between correctness of two classification models, with McNemar test, we need the flags of answer correctness for each element of the test data. The collection of correctness flags can be obtained by adding to the configuration of classifiers tests the correctness commentator (*CorrectnessCommentator*) as it was presented in section 3 in the paragraph devoted to commentators. Additionally, the main label should be set to the "Correctness". Declarations:

```
s1 = knnSeries.Filter("Repetition", 1).Filter("Fold", 1);
s2 = svmSeries.Filter("Repetition", 1).Filter("Fold", 1);
```

select the results for particular models. By double filtering (superposition), models belonging to the first repetition and the first fold of CV are selected independently for kNN and SVM test series. Because the correctness label-value item contains a collection of correctness for each vector the *Unpack()* method must be used (it converts the collection of collections into one collection):

```
s1 = s1.Unpack();    s2 = s2.Unpack();
```

Now series *s1* and *s2* contain the correctness for selected models and are ready to be used via McNemar test:

```
s = McNemar.Transform(s1, s2);
```

The final series *s* contains test results: the p-value and McNemar statistic value, accessible via *s["p-value"]* and *s["statistic"]* respectively.

In even simpler way the McNemar test can be applied to all subsequent tests of the CV. In this case the series of collections of correctness is also unpacked to one flat collection (result of ll 4 test in a single series) and then put to the McNemar test:

```
s = McNemar.Transform(knnSeries.Unpack(), svmSeries.Unpack());
```

*Basic statistics.* When collecting *knnSeries* and *svmSeries* as it was presented at the beginning of this section with *MainLabel* set to "Accuracy", the basic statistics can be directly computed:

```
s = knnSeries.Transform(new BasicStatistics());
```

Indexing series  $s$  with "Minimum", "Mean", "Maximum", "Standard deviation" respective properties are captured (for example  $s["Mean"]$ ). The *BasicStatistics* transformer sets the "Mean" as the main label.

Using grouping, the independent statistics for each CV test can be computed. To do this first we have to group the series by the repetition index:

```
 $s1 = knnSeries.Group("Repetition");$ 
```

After that, the  $s1$  series contains subseries with items representing subsequent CV tests. Now, the basic statistics transformer can be applied to the series:

```
 $s1 = s1.MAP(new BasicStatistics());$ 
```

The *MAP* transformer performs the transformation given as the parameter (here *BasicStatistics*) on each of the subseries and the results are collected into a new series. Note that the *MAP* in general can be nested if needed. *MAP* may also be very useful with transformations like grouping, ungrouping, filtering and many others. Finally the  $s1$  series contains a sequence of subseries with basic statistics of CV repetitions and by calling the ungroup transformer we obtain series of statistics of CV's (in the main series not in subseries):

```
 $s1 = s1.Ungroup();$ 
```

Now  $s1$  contains items, one per single CV, and each one contains mean, minimum, maximum, variance and standard deviation. All these operations may be composed into a single transformation chain:

```
 $s1 = knnSeries.Group("Repetition").MAP(new BasicStatistics()).Ungroup();$ 
```

giving exactly the same result.

Now, the following code computes the statistics of inner CV test statistics:

```
 $s1 = s1.Transform(new BasicStatistics());$ 
```

After that  $s1["Mean"]$  represents average accuracy and  $s1["Standard deviation"]$  represents standard deviation of the mean accuracies of the repetitions of CV tests.

### *T-test and paired t-test*

To test the statistical significance of the differences between the results of kNN and SVM with paired t-test (it does not make much sense in the case of  $2 \times 2$ -fold CV, but the way to do it does not depend on the numbers of repetitions or CV folds), we need to collect the results of all the kNN models and SVM models separately in the same data distributions, as it was already presented at the beginning of this section with the main label set up to "Accuracy".

We can directly compute statistical significance between accuracies of kNN and SVM calling paired t-test as follows:

```
 $s = TTestPaired(knnSeries, svmSeries);$ 
```

The  $s$  series can be indexed by "p-value" or "statistic" to get the most interesting values calculated by the t-test ( $s["statistic"]$  is the value of t). In the case when

knnSeries and svmSeries are not collected from the same CV test, and we are interested in calculating t-test, we just substitute *TTestPaired* by *TTest* in the call presented above.

If we want to compare mean accuracy and stability of two methods, in the analysis we can replace mean accuracies with the differences between means and standard deviations of accuracies:  $\overline{acc} - \alpha \sigma_{acc}$  where the  $\alpha$  is a parameter (typically equal to 1). If we are also interested in the t-test on differences between such measures, we may call it in the following way:

```
s = TTestPaired.Transform(s1 - s1.GetSeries("Standard deviation"),
    s2 - s2.GetSeries("Standard deviation"));
```

First, the series of differences are calculated (*s1 - s1.GetSeries("Standard deviation")*), and the same for *s2*). Note that the transformation *GetSeries* creates new series with "Standard deviation" as the main label and containing the same items as *s1* (the main label of *s1* is "Mean").

To call the Wilcoxon signed rank test or Mann-Whitney test all we need is the substitution of *TTestPaired* or *TTest* by *Wilcoxon* and *Mann-Whitney* respectively, in the above examples.

The number of possible combinations of different commentators, queries and series grouping, filtering and transformations is huge even with a small set of basic commentators and transformations. The code that must be written is short and intuitive. Since the system is open in the sense that any SDK user can add new commentators and series transformations, the possibilities of results analysis are so rich, that we can claim, they are restricted only by user invention.

## 6 Summary

There is no meta-learning without meta-knowledge. The system of result repository and queries has been designed to facilitate advanced meta-learning. It can be successfully used for miscellaneous, sophisticated applications in data mining including construction of different types of classification committees and other ensembles of models, feature selection and extraction, and many other fields.

Universal mechanisms for results repository services and powerful system of repository information retrieval and manipulation, offers incomparable possibilities never met in known meta-learning or other data mining systems. The results repository may contain heterogeneous information. Moreover the commentators may be used to extract additional information from already implemented machines to extend possibilities of further analysis. Using the system of results repository queries, series and series transformers, one can easily obtain answers for very broad range of questions and successfully mine for meta-knowledge.

**Acknowledgements.** The research is supported by the Polish Ministry of Science with a grant for years 2005–2007.

## References

1. Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: Meta-learning by landmarking various learning algorithms. In: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann (June 2000) 743–750
2. Brazdil, P., Soares, C., da Costa, J.P.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning* **50**(3) (2003) 251–277
3. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature extraction, foundations and applications. Springer (2006)
4. Jankowski, N., Grąbczewski, K.: Heterogenous committees with competence analysis. In Nedjah, N., Mourelle, L., Vellasco, M., Abraham, A., Köppen, M., eds.: Fifth International conference on Hybrid Intelligent Systems, Brasil, Rio de Janeiro, IEEE, Computer Society (November 2005) 417–422
5. Grąbczewski, K., Jankowski, N.: Versatile and efficient meta-learning architecture: Knowledge representation and management in computational intelligence. In: IEEE Symposium Series on Computational Intelligence (SSCI 2007), IEEE (2007) 51–58

---

# Maximum Margin Classification on Convex Euclidean Metric Spaces

André Stuhlsatz<sup>1,2</sup>, Hans-Günter Meier<sup>2</sup>, and Andreas Wendemuth<sup>1</sup>

<sup>1</sup> Otto-von-Guericke University Magdeburg, Germany, Cognitive Systems Group, Department of Electrical Engineering and Information Technology

<sup>2</sup> University of Applied Sciences Düsseldorf, Germany, Department of Electrical Engineering

`andre.stuhlsatz@fh-duesseldorf.de`

**Summary.** In this paper, we present a new implementable learning algorithm for the general nonlinear binary classification problem. The suggested algorithm abides the maximum margin philosophy, and learns a decision function from the set of all finite linear combinations of continuous differentiable basis functions. This enables the use of a much more flexible function class than the one usually employed by Mercer-restricted kernel machines. Experiments on 2-dimensional randomly generated data are given to compare the algorithm to a Support Vector Machine. While the performances are comparable in case of Gaussian basis functions and static feature vectors the algorithm opens a novel way to hitherto intractable problems. This includes especially classification of feature vector streams, or features with dynamically varying dimensions as such in DNA analysis, natural speech or motion image recognition.

## 1 Overcoming Mercer Conditions

In the late seventies, Vapnik and Chervonenkis [1] had introduced the concept of a maximum margin separating hyperplane. It has been shown, that the maximization of the margin between the data points and the hyperplane minimizes an upper bound on the actual risk [2]. This strategy often leads to a better generalization performance in comparison to pure empirical approaches. E.g., the Support Vector Machine (SVM) [2] has been proved to provide very good classification performance on many different applications in comparison to other learning paradigms [3]. Although the separating hyperplane realizes a linear classification rule, it is also possible to obtain nonlinear ones using decision functions consisting of nonlinear basis functions called kernels. Unfortunately kernel functions have to satisfy some non-trivial mathematical conditions (Mercer conditions [4]). The Mercer conditions are proved only for a handful of commonly used kernels, see e.g. [5]. Sometimes one drawback can be the lack of variety of feasible kernels in applications where the geometries of the data space are not well modeled by the given kernels or where we have streams of features of variable length instead of static feature vectors. For example, this is the case in speech recognition, motion image recognition or DNA sequence analysis. The aim of this paper is to provide a novel approach for a solution to these cases. We



will present a practicable method of maximum margin classification in Banach spaces based on the theoretical work of [6]. Our algorithm will learn on arbitrary compact and convex euclidean metric spaces and its solution will be a decision function from the set of all finite linear combinations of continuous differentiable basis functions.

## 2 Lipschitz Classifier

Usually, nonlinear classifiers implement decision functions<sup>1</sup> of a function set

$$\mathcal{F} := \left\{ f : \mathcal{X} \rightarrow \mathbb{R} : f(\mathbf{x}) := \sum_{n=1}^M c_n \Phi_n(\mathbf{x}) + b, \quad c_n, b \in \mathbb{R}, M \in \mathbb{N} \right\} \quad (1)$$

with linearly weighted basis functions  $\Phi_n : \mathcal{X} \rightarrow \mathbb{R}$ . E.g., in case of the SVM,  $\Phi_n(\mathbf{x}) := k(\mathbf{x}_n, \mathbf{x})$ , where  $k$  is some kernel function which has to satisfy the Mercer conditions [4] and  $\mathcal{X}$  is a vector space equipped with some scalar product (Hilbert space). Popular kernels are the Gaussian kernel function  $k(\mathbf{x}_n, \mathbf{x}) := \exp(-\gamma \|\mathbf{x}_n - \mathbf{x}\|_2^2)$  or the polynomial kernel  $k(\mathbf{x}_n, \mathbf{x}) := \langle \mathbf{x}_n, \mathbf{x} \rangle_2^d$ . The kernel function  $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}$  represents a scalar product of an appropriate space  $\mathcal{H}$  after embedding  $\mathcal{X}$  into  $\mathcal{H}$  by some feature map  $\phi$ . In particular, the SVM seeks for an  $f \in \mathcal{F}$  so that the empirical error is minimized and the margin  $\rho = 1/\|f\|_{\mathcal{H}}$  between the data points and a separating hyperplane in  $\mathcal{H}$  is maximized. The main concept [6] of our algorithm is based upon the idea of embedding a bounded metric space  $(\mathcal{X}, d)$ , i.e.  $\text{diam}(\mathcal{X}) := \sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} d(\mathbf{x}, \mathbf{y})$  which is finite with some metric  $d$ , isometrically into a special Banach space  $\mathcal{B}$  (Arens-Eells space). The corresponding function space  $\mathcal{Lip}(\mathcal{X})$  of bounded Lipschitz continuous functions is isometrically mapped to the dual space  $\mathcal{B}'$ . Recall a function is Lipschitz, if  $L(f) := \sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X}, \mathbf{x} \neq \mathbf{y}} \frac{|f(\mathbf{x}) - f(\mathbf{y})|}{d(\mathbf{x}, \mathbf{y})} < \infty$  exists.  $L(f)$  is defined as the smallest Lipschitz constant of  $f$ . In analogy to the SVM embedding, it is possible to assign  $f \in \mathcal{Lip}(\mathcal{X})$  to a linear functional  $T_f(m_{\mathbf{x}}) = f(\mathbf{x})$  using appropriate representations  $m_{\mathbf{x}} \in \mathcal{B}$  of the points  $\mathbf{x} \in \mathcal{X}$ , such that  $\|T_f\| = L(f)$ . A canonical hyperplane  $H := \{m_{\mathbf{x}} \in \mathcal{B} \mid T_f(m_{\mathbf{x}}) = 0\}$  in  $\mathcal{B}$  allows to bound the margin  $\rho$  between the mapped data  $m_{\mathbf{x}_n}$  and  $H$  from below by  $\rho \geq 1/\|T_f\| = 1/L(f)$ . Note that, contrary to the SVM approach, minimizing  $\|T_f\| = L(f)$  maximizes a lower bound of the margin  $\rho$ . As a consequence, a large margin algorithm can be constructed on a metric space by picking up a function with small Lipschitz constant and minimum empirical error from a larger set of decision functions  $\mathcal{F} \cap \mathcal{Lip}(\mathcal{X})$  than in case of the SVM. Using the standard methodology to handle non-separable data as well [2], the Lipschitz classifier yields a constrained optimization problem of the form

$$\inf_{(f, \xi) \in \mathcal{F} \cap \mathcal{Lip}(\mathcal{X}) \times [0, \infty)^N} L(f) + C \sum_{n=1}^N \xi_n \quad \text{w.r.t.} \quad y_n f(\mathbf{x}_n) \geq 1 - \xi_n \quad \forall n \in I \quad (2)$$

---

<sup>1</sup> Note, that we associate the term decision function with  $f \in \mathcal{F}$  instead of  $x \mapsto \text{sign}(f(\mathbf{x}))$ .

with index set  $I := \{1, \dots, N\}$ , slack-variable  $\boldsymbol{\xi} \in [0, \infty)^N$ , training data  $(\mathbf{x}_n, y_n) \in \mathcal{X} \times \{\pm 1\}$  and parameter  $C > 0$  free but fixed in advance.

### 3 Learning over Compact and Convex Metric Spaces

As long as no additional assumptions restrict the decision function space  $\mathcal{F}$  and the underlying metric space  $\mathcal{X}$ , it has been shown in [6] how to solve the Lipschitz classification problem, explicitly. The problem was solved as a Lipschitz minimal norm interpolation problem over the whole Lipschitz function space  $\mathcal{Lip}(\mathcal{X})$  using the notion of Lipschitz extension. Usually, a solution found by this procedure may be expected to be overfitted, because of the high complexity of the considered function space. An approximation is obtainable if the function class is restricted to linear combinations of distances and additionally assuming the distance as the metric on  $\mathcal{X}$  [7]. Instead of an approximation or the Lipschitz extension interpolant, it is possible by the following theorem to determine exact Lipschitz constants of real valued, continuous differentiable decision functions defined on a compact and convex euclidean space (for a proof see appendix).

**Theorem 1.** *Suppose, that  $\mathcal{V} := (\mathbb{R}^m, \|\cdot\|_2)$  is a linear space,  $\mathcal{D} \subseteq \mathcal{V}$  open and  $\mathcal{X} \subset \mathcal{D}$  compact and convex. If  $f : \mathcal{D} \rightarrow \mathbb{R}$  is continuous differentiable, then*

$$L(f) = \max_{\mathbf{x} \in \mathcal{X}} \|\nabla f(\mathbf{x})\|_2. \quad (3)$$

Hence, equation (3) provides an opportunity to reformulate problem (2) by substituting  $L(f)$  with  $\max_{\mathbf{x} \in \mathcal{X}} \|\nabla f(\mathbf{x})\|_2$ , where  $\mathcal{X} \subset \mathbb{R}^m$  is compact and convex, and minimizing over  $\mathcal{F} \cap \mathcal{C}^{(1)}(\mathcal{X}, \mathbb{R}) \subset \mathcal{Lip}(\mathcal{X})$ . In the following, we will consider a function space consisting of finite linear combinations of continuous differentiable basis functions.

**Lemma 1.** *Suppose  $\mathcal{D} \subseteq \mathbb{R}^m$  is open,  $\mathcal{X} \subset \mathcal{D}$  is compact and convex,  $f \in \mathcal{F}$  as defined in (1), every  $\Phi_n : \mathcal{D} \rightarrow \mathbb{R}$  is a continuous differentiable basis function. Then  $\mathcal{F} \subset \mathcal{Lip}(\mathcal{X})$  and the Lipschitz classification problem (2) can be stated as*

$$\min_{(\mathbf{c}, \boldsymbol{\xi}, b) \in \mathcal{Z}} \max_{\mathbf{x} \in \mathcal{X}} \frac{1}{2} \mathbf{c}^T \mathbf{K}(\mathbf{x}) \mathbf{c} + C \mathbf{1}_N^T \boldsymbol{\xi} \quad (4)$$

$$\mathcal{Z} := \left\{ (\mathbf{c}, \boldsymbol{\xi}, b) \in \mathbb{R}^{M+N+1} : \boldsymbol{\xi} \geq \mathbf{0}, \mathbf{Y} \mathbf{G} \mathbf{c} + \mathbf{y} b - \mathbf{1}_N + \boldsymbol{\xi} \geq \mathbf{0} \right\},$$

where  $\mathbf{K}(\mathbf{x}) \in \mathbb{R}^{M \times M}$  is a symmetric positive semi-definite matrix with  $\mathbf{K}(\mathbf{x})_{m,n} := \langle \nabla \Phi_m(\mathbf{x}), \nabla \Phi_n(\mathbf{x}) \rangle_2$ ,  $\mathbf{G} \in \mathbb{R}^{N \times M}$  the data dependent design matrix with  $\mathbf{G}_{n,m} := \Phi_m(\mathbf{x}_n)$ ,  $\mathbf{Y} := \text{diag}(\mathbf{y}) \in \mathbb{R}^{N \times N}$  a diagonal matrix of a given target vector  $\mathbf{y} \in \{-1, 1\}^N$  with components  $y_n$ ,  $\boldsymbol{\xi} \in \mathbb{R}^N$  the slack-variable of the soft margin formulation with components  $\xi_n$  and  $\mathbf{1}_N := (1, \dots, 1)^T \in \mathbb{R}^N$  the vector of ones.

Optimization problem (4) is equivalent to (2) in the sense, that from a solution  $(\mathbf{c}^*, \boldsymbol{\xi}^*, b^*, \mathbf{x}^*)$  of (4) follows a solution  $(f^*, \boldsymbol{\xi}^*)$  of (2) with  $f^*(\mathbf{x}) := \sum_{n=1}^M c_n^* \Phi_n(\mathbf{x}) + b^*$ . It can be easily verified a maximizer of  $\max_{\mathbf{x} \in \mathcal{X}} \|\nabla f(\mathbf{x})\|_2$

is also a maximizer of  $\max_{\mathbf{x} \in \mathcal{X}} \|\nabla f(\mathbf{x})\|_2^2 / 2$ . The matrix  $\mathbf{K}(\mathbf{x})$  is semi-positive definite for all  $\mathbf{x} \in \mathcal{X}$ , because

$$\|\nabla f(\mathbf{x})\|_2^2 = \sum_{m=1}^M \sum_{n=1}^M c_m c_n \langle \nabla \Phi_m(\mathbf{x}), \nabla \Phi_n(\mathbf{x}) \rangle_2 = \mathbf{c}^T \mathbf{K}(\mathbf{x}) \mathbf{c} \quad (5)$$

implies  $\mathbf{c}^T \mathbf{K}(\mathbf{x}) \mathbf{c} \geq 0 \forall \mathbf{c} \in \mathbb{R}^M$ . As the pointwise maximum of a family of convex functions,  $\max_{\mathbf{x} \in \mathcal{X}} \mathbf{c}^T \mathbf{K}(\mathbf{x}) \mathbf{c}$  is convex in  $\mathbf{c} \in \mathbb{R}^M$  as well.

### 3.1 Lipschitz Classifier Reformulation as Dual SIP

In the previous section we obtained a constrained minimax problem (4) which is not easily solvable directly, because of the complex feasible set as well as the inner global maximization dependent on the outer minimization and vice versa. But it is known that this minimax problem can be transformed into a equivalent problem called semi-infinite program (SIP). Semi-infinite programming deals with problems of infinitely many constraints in finitely many variables. To reveal a brief idea, we state following well-known proposition from the theory of convex SIP without proof:

**Proposition 1.** *Suppose  $\mathcal{X} \subseteq \mathbb{R}^m$  to be compact,  $\mathcal{Y} \subseteq \mathbb{R}^n$  to be closed and convex, as well as  $g: \mathcal{Y} \times \mathcal{X} \rightarrow \mathbb{R}$  continuous with convex functions  $g(\cdot, \mathbf{x})$ . Then each solution of the convex semi-infinite program*

$$(\mathbf{c}_0^*, \mathbf{c}^*) := \underset{(\mathbf{c}_0, \mathbf{c}) \in \mathcal{Z}_{SIP}}{\operatorname{argmin}} \ c_0 \quad (6)$$

with feasible set  $\mathcal{Z}_{SIP} := \{(\mathbf{c}_0, \mathbf{c}) \in \mathbb{R} \times \mathcal{Y} : g(\mathbf{c}, \mathbf{x}) - c_0 \leq 0 \forall \mathbf{x} \in \mathcal{X}\}$  admits a solution  $(\mathbf{c}^*, \mathbf{x}^*) \in \mathcal{Y} \times \mathcal{X}$  of the problem

$$c_0^* = g(\mathbf{c}^*, \mathbf{x}^*) = \min_{\mathbf{c} \in \mathcal{Y}} \max_{\mathbf{x} \in \mathcal{X}} g(\mathbf{c}, \mathbf{x}). \quad (7)$$

For further particulars we refer you to [8] and references therein. Applying above SIP reformulation as well as duality results for convex SIP [9] yield the semi-infinite dual of the minimax problem (4) of the Lipschitz classifier:

$$(D) \quad \max_{\mathbf{K} \in \operatorname{conv}(\mathcal{K})} q(\mathbf{K}) \quad (8)$$

where  $\mathcal{K} := \{\mathbf{K}(\mathbf{x}) \succeq 0 : \mathbf{x} \in \mathcal{X}\}$  collects the semi-positive definite matrices on  $\mathcal{X}$ ,

$$q(\mathbf{K}) := \max_{(\boldsymbol{\alpha}, \mathbf{c}) \in \mathcal{Z}_q(\mathbf{K})} Q(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{K}), \quad Q(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{K}) := \boldsymbol{\alpha}^T \mathbf{1}_N - \frac{1}{2} \mathbf{c}^T \mathbf{K} \mathbf{c}, \quad (9)$$

$$\mathcal{Z}_q(\mathbf{K}) := \left\{ (\boldsymbol{\alpha}, \mathbf{c}) \in \mathbb{R}^{N \times M} : 0 \leq \boldsymbol{\alpha} \leq C \mathbf{1}_N, \boldsymbol{\alpha}^T \mathbf{y} = 0, \mathbf{K} \mathbf{c} = \mathbf{G}^T \mathbf{Y} \boldsymbol{\alpha} \right\}.$$

---

**Algorithm 1.** Lipschitz classifier

---

**Input:**  $\mathbf{K}^{(1)} := \mathbf{K}(\mathbf{x}_0) \in \mathcal{K}$ ,  $T$ **Set**  $t = 0$ .**repeat**1. Set  $t=t+1$ .

2. Calculate

$$(\boldsymbol{\alpha}^{(t)}, \mathbf{c}^{(t)}) := \underset{(\boldsymbol{\alpha}, \mathbf{c}) \in \mathcal{Z}_q(\mathbf{K}^{(t)})}{\operatorname{argmax}} Q(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{K}^{(t)}).$$

3. Search for  $\mathbf{x} \in \mathcal{X}$  such that  $\mathbf{K}(\mathbf{x}) \in \mathcal{K}$  and

$$Q(\boldsymbol{\alpha}^{(t)}, \mathbf{c}^{(t)}, \mathbf{K}(\mathbf{x})) > Q(\boldsymbol{\alpha}^{(t)}, \mathbf{c}^{(t)}, \mathbf{K}^{(t)})$$

else, if such  $\mathbf{x}$  does not exist terminate.

4. Calculate

$$\lambda^* := \underset{\lambda \in [0,1]}{\operatorname{argmax}} q\left(\lambda \mathbf{K}(\mathbf{x}) + (1 - \lambda) \mathbf{K}^{(t)}\right)$$

5. Set  $\mathbf{K}^{(t+1)} = \lambda^* \mathbf{K}(\mathbf{x}) + (1 - \lambda^*) \mathbf{K}^{(t)}$ .**until**  $t=T$ 

---

The closure of the convex hull of  $\mathcal{K}$  is denoted as  $\overline{\operatorname{conv}(\mathcal{K})}$  while the convex hull is the intersection of all convex sets containing  $\mathcal{K}$ . Because of the difficulties in the computation of the derivatives of  $\mathbf{K}(\mathbf{x})$ , we applied a two stage greedy strategy giving rise to the algorithm (1). In each iteration step, the algorithm finds a solution of (8) on a more and more growing convex subset of the convex hull  $\overline{\operatorname{conv}(\mathcal{K})}$  which is spanned by a sequence of matrices  $\mathbf{K}^{(t)} \in \mathcal{K}$  such that  $q(\mathbf{K}^{(1)}) < q(\mathbf{K}^{(2)}) < \dots < q(\mathbf{K}^{(T)})$ . Therefore, one has to search for a  $\mathbf{x} \in \mathcal{X}$  so that  $Q(\boldsymbol{\alpha}^{(t)}, \mathbf{c}^{(t)}, \mathbf{K}(\mathbf{x})) > Q(\boldsymbol{\alpha}^{(t)}, \mathbf{c}^{(t)}, \mathbf{K}^{(t)}) = q(\mathbf{K}^{(t)})$ . Using this new candidate  $\mathbf{K}(\mathbf{x}) \in \mathcal{K}$  we conclude  $q(\mathbf{K}(\mathbf{x})) > q(\mathbf{K}^{(t)})$ , thus the algorithm will calculate the new optimal convex combination of the given matrices  $\mathbf{K}(\mathbf{x})$  and  $\mathbf{K}^{(t)}$  in the sense that it will improve the next iterate

$$q(\mathbf{K}^{(t+1)}) = \max_{\lambda \in [0,1]} q\left(\lambda \mathbf{K}(\mathbf{x}) + (1 - \lambda) \mathbf{K}^{(t)}\right) \geq q(\mathbf{K}(\mathbf{x})) > q(\mathbf{K}^{(t)}). \quad (10)$$

On the other hand, if no  $\mathbf{x} \in \mathcal{X}$  can be found to improve the next iterate or the maximum number of steps are reached, the algorithm terminates. After obtaining a solution  $(\boldsymbol{\alpha}^*, \mathbf{K}^*)$  one gets the associated primal solution  $(\mathbf{c}^*, b^*)$  as a solution of the equations  $\mathbf{K}^* \mathbf{c}^* = \mathbf{G}^T \mathbf{Y} \boldsymbol{\alpha}^*$  and  $b^* = -\frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} c_i^* \Phi(\mathbf{x}_i)$ . The index set  $\mathcal{S}$  contains all indices  $i$  with  $0 < \alpha_i < C$ . The former equations are easily obtained from the derivation of the dual problem (8).

## 4 Experiments

We investigated an artificial, 2-dimensional and randomly generated classification problem for evaluation, training and testing. A total of one evaluation set, 10 training sets and 10 test sets from a mixture of Gaussians were generated. All sets were pairwise disjoint. First we simulated for the positive labeled class 10 means  $\mu_k$  from a bivariate Gaussian distribution  $N((1, 0)^T, \mathbf{I})$ . Similarly, for

the negative labeled class 10 further means were drawn from  $N((0, 1)^T, \mathbf{I})$ . Then for each class 50 vectors for evaluation, 100 vectors for each of the training sets and 200 vectors for each of the test sets were generated as follows: we picked a  $\mu_k$  by chance with probability 1/10, and then each observation was generated from a further Gaussian  $N(\mu_k, \mathbf{I}/5)$ . This procedure of generating the data is taken out of [10]. For comparison, a Support Vector Machine was investigated on the same data as well as the Lipschitz classifier. In both classifiers, we employed Gaussian kernels  $\Phi_n(\mathbf{x}) := \exp(-\gamma\|\mathbf{x}_n - \mathbf{x}\|_2^2)$  as basis functions to get comparable and better interpretable results. Before the test experiments have been started, we had to adjust all parameters of the learning machines, e.g. the algorithm parameter  $C$ , the kernel parameter  $\gamma$  and the tuning parameters of the simulated annealing method [11] we used in step 3. of the algorithm. Evaluation was done independent of the test sets using one training set in conjunction with the evaluation set. Next, on every training set a Lipschitz classifier and a SVM were trained with fixed parameter settings from the evaluation (Lipschitz classifier:  $C = 100, \gamma = 0.095$ , SVM:  $C = 80, \gamma = 0.35$ ) resulting in 10 different classifiers for each learning machine. For testing the Lipschitz classifier as well as the SVM, we performed a recognition run on every test set using one of the trained classifiers. This means for example, that the first trained SVM was tested on the first generated test set. Similarly, the second trained SVM was tested on the second generated test set, and so on. From this procedure 10 test results form the Lipschitz classifier and SVM have been obtained. These results are summarized in table (1).

**Table 1.** Classification accuracies for the Lipschitz classifier and Support Vector Machine on ten 2-dimensional randomly generated test sets. Each test set contains 400 data points.

#Test	SVM [%]	Lipschitz classifier [%]
1	<b>87.25</b>	85.50
2	<b>87.50</b>	86.00
3	90.75	<b>91.25</b>
4	<b>85.75</b>	84.25
5	88.50	<b>88.75</b>
6	86.00	<b>86.25</b>
7	<b>91.00</b>	89.00
8	<b>83.50</b>	83.00
9	<b>88.00</b>	87.75
10	<b>86.50</b>	85.50
mean	87.48±2.27	86.73± 2.46

## 5 Conclusion and Future Work

As shown in table (1), the achieved freedom to use arbitrary continuous differentiable basis functions provided by our novel implementation of a Lipschitz

classifier leads to a small relative loss of 0.86%, only. This suggests that the algorithm can be used in any application where a kernel based classifier like the SVM is not directly capable. Using the theory of a Banach space embedding, we justified that our novel algorithm is indeed a maximum margin classifier, thus it has the potential to generalize well as the results of our experiments indicate, too. The given results will focus our future work on using other basis functions than Mercer kernels and on applying our algorithm to applications demanding streams of feature vectors with variable length. Such experiments using data from DNA analysis or motion image recognition will enable us to investigate the performance of the Lipschitz algorithm on real world data.

## References

1. Vapnik V, Chervonenkis A (1979) Estimation of dependences based on empirical data. [in russian]
2. Vapnik V (1995) The nature of statistical learning theory. Springer, Berlin Heidelberg New York
3. Stuhlsatz A, Katz M, Krüger SE, Meier HG, Wendemuth A (2006) Support vector machines for postprocessing of speech recognition hypotheses. Proceedings of International Conference on Telecommunications and Multimedia TEMU
4. Courant R, Hilbert D (1953) Methods of mathematical physics. Interscience Publishers, Inc.
5. Schölkopf B, Smola AJ (2002) Learning with kernels. The MIT Press.
6. von Luxburg U, Bousquet O (2004) Distance based classification with lipschitz functions. Journal of Machine Learning Research, 5: 669-695
7. Graepel T, Herbisch R, Schölkopf B, Smola AJ, Bartlett P, Müller KR, Obermayer K, Williams R (1999) Classification of proximity data with LP machines. Proceedings of 9th International Conference on Artificial Neural Networks: 304-309
8. Hettich R, Kortanek KO (1993) Semi-infinite programming: Theory, methods and applications. SIAM Reviews, 35: 380-429
9. Shapiro A (2005) On the duality theory of convex semi-infinite programming. Optimization, 54: 535-543
10. Hastie T, Tibshirani R, Friedman J (2003) The elements of statistical learning. Springer, Berlin Heidelberg New York
11. Liu JS (2001) Monte Carlo strategies in scientific computing. Springer, Berlin Heidelberg New York

## A Appendix

**Proof of theorem 1.** *If  $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ ,  $\mathbf{x}_1 \neq \mathbf{x}_2$  and*

*$\mathcal{S} := \{\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 : \lambda \in (0, 1)\} \subseteq \mathcal{X}$ , then by the Mean-Value-Theorem there exists  $\hat{\mathbf{x}} \in \mathcal{S}$ , so that*

$$\begin{aligned} |f(\mathbf{x}_1) - f(\mathbf{x}_2)| &= \|\nabla f(\hat{\mathbf{x}})^T(\mathbf{x}_1 - \mathbf{x}_2)\|_2 \leq \|\nabla f(\hat{\mathbf{x}})\|_2 \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \\ \Leftrightarrow \frac{|f(\mathbf{x}_1) - f(\mathbf{x}_2)|}{\|\mathbf{x}_1 - \mathbf{x}_2\|_2} &\leq \|\nabla f(\hat{\mathbf{x}})\|_2 \leq \max_{\mathbf{x} \in \mathcal{X}} \|\nabla f(\mathbf{x})\|_2 \end{aligned}$$

$$\Rightarrow L(f) = \max_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}, \mathbf{x}_1 \neq \mathbf{x}_2} \frac{|f(\mathbf{x}_1) - f(\mathbf{x}_2)|}{\|\mathbf{x}_1 - \mathbf{x}_2\|_2} \leq \max_{\mathbf{x} \in \mathcal{X}} \|\nabla f(\mathbf{x})\|_2.$$

For  $\mathbf{x} \in \mathcal{X}$  and some  $\mathbf{v} \in \mathbb{R}^m$  with  $\|\mathbf{v}\|_2 = 1$ , the directional derivative reads as

$$\lim_{t \rightarrow 0} \left( \frac{f(\mathbf{x} + t\mathbf{v}) - f(\mathbf{x})}{t} \right) = \partial_{\mathbf{v}} f(\mathbf{x}) = \nabla f(\mathbf{x})^T \mathbf{v}.$$

Next, we conclude for  $\|\nabla f(\mathbf{x})\|_2 \neq 0$  and  $\mathbf{v} = \nabla f(\mathbf{x}) / \|\nabla f(\mathbf{x})\|_2$ , that

$$\|\nabla f(\mathbf{x})\|_2 = \lim_{t \rightarrow 0} \frac{|f(\mathbf{x} + t\mathbf{v}) - f(\mathbf{x})|}{|t|} \leq L(f).$$

It follows the lower bound

$$\max_{\mathbf{x} \in \mathcal{X}} \|\nabla f(\mathbf{x})\|_2 \leq L(f). \quad \square$$

---

# Optimization of the SVM Screening Kernel— Application to Hit Definition in Compound Screening

Karol Kozak<sup>1</sup>, Marta Kozak<sup>1</sup>, and Katarzyna Stapor<sup>2</sup>

<sup>1</sup> Max Planck Institute of Molecular Cell Biology, 01307 Dresden, Germany  
kozak@mpi-cbg.de

<sup>2</sup> Silesian Technical University, 44-100 Gliwice, Poland  
katarzyna.stapor@polsl.pl

**Summary.** This study applies the Support Vector Machine (SVM) algorithm to the problem of chemical compound screening with a desired activity and hit definition. The problem of automatically tuning multiple parameters for pattern recognition SVMs using our new introduced kernel for chemical compounds is considered. This is done by a simple eigen analysis method which is applied to the matrix of the same dimension as the kernel matrix to find the structure of feature data, and to find the kernel parameter accordingly. We characterize distribution of data by the principle component analysis method.

## 1 Introduction

Biological screening processes are used in drug discovery to screen large numbers of compounds against a biological target. By quantifying structure-activity relationship (QSAR) means a group of bioinformatics or chemoinformatics algorithms, typically a using physico-chemical parameters or three-dimensional molecular fields with statistical techniques. Finding active compounds amongst those screened, QSAR algorithms are aimed at planning additional active compounds without having to screen each of them individually. Classically, QSAR models are designed by representing molecules by a large set of descriptors, i.e. by a high dimensional vector, and then applying some kind of Pattern Recognition methods. One of the most important design choices for SVMs is the kernel-parameter, which implicitly defines the structure of the high dimensional feature space where a maximal margin hyperplane will be found. However, before this stage is reached in the use of SVMs, the actual kernel must be chosen. Our previous work [5] focused on graph-structured data on screening compounds and on the development of kernel methods for graph-structured data in combination with molecular descriptors, with particular emphasis on QSAR in the screening process and the prediction of the biological activity of chemical compounds. Model selection of the SVM involves two hyperparameters: the penalty parameter  $C$  and the kernel parameter used in the Gaussian kernel. The selection of suitable parameters for Gaussian kernel used in proposed screening kernel



is crucial for effective classification. In this paper we analyze the distribution of data in the feature space on the basis of the principal component analysis (PCA) method, and propose a simple method that predicts a good set of Gaussian kernel parameter quickly for a particular problem domain in chemical compounds.

## 2 Support Vector Machine Classification

Let us use a training data set  $\{m_i; \omega_i\}; i = 1, \dots, l$  where  $\omega_i \in \{-1, 1\}$  represents the label of an arbitrary example  $\omega_i$ ,  $R_d$ ,  $d$  being the dimension of the input space. Also, let us define a linear decision surface by the equation  $f(x) = w : m + b = 0$ . SVMs exactly implement the idea of regularized risk minimization by maximizing the margin between the two classes. This can be achieved through the minimization of  $\|w\|^2$ . The solution of which is:

$$w = \sum_{i=1}^l \alpha_i \omega_i m_i \quad (1)$$

with the constraint  $w = \sum_i \alpha_i \omega_i = 0$ . The parameters  $\alpha_i$  are found by maximizing the dual objective [10][11]

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \omega_i \cdot \omega_j m_i m_j \quad (2)$$

However, in real-life classification problems, the algorithm as stated above is unable to achieve perfect separation between two classes especially in case of noisy data. One key aspect of the SVM model besides a well optimized C penalization parameter called a trade-off parameter is that the data enters the above expressions only in the form of the dot product of pairs. This leads to the resolution of the second problem mentioned above, namely that of non-linearly separable data. The basic idea with SVMs is to map the training data into a higher dimensional feature space via some mapping  $\phi(m)$  and construct a separating hyperplane with maximum margin there. This yields a nonlinear decision boundary in the original input space. The kernel can deal in a uniform way with a multitude of data types and can be used to detect many types of relations in data. We assume that  $m$  is mapped to  $p$  dimensional kernel feature space  $F$  by  $\phi(m) \in F$   $m = (m_1, \dots, m_n)^t \rightarrow \phi(m) = (\phi_1(m), \dots, \phi_p(m))^t$ . Kernel methods computes inner product  $K(m, m') = \langle \phi(m) \cdot \phi(m') \rangle = \phi(m)^t \cdot \phi(m')$ , without referring the feature  $\phi_i (i = 1, \dots, p)$  directly. Let  $K$  be  $K_{ij} = \langle \phi(m_i) \cdot \phi(m'_j) \rangle$  then  $K = \phi^t \phi$  with  $\phi = (\phi(x_1)' \dots, \phi(x_l))$  (sample matrix). Typical choices for kernels: linear, sigmoid, polynomial, RBF, gaussian.

Because chemical compounds are often represented by the graph of their covalent bonds, pattern recognition methods in this domain must be capable of processing graphical structures with variable size. Let us assume now we have two molecules  $m$  and  $m'$ , which have atoms  $a_1, \dots, a_{|m|}$  and  $a'_1, \dots, a'_{|m'|}$ . Let us further assume we have a kernel  $k_{nei}$ , which compares a pair of atoms  $(a_h, a'_{h'})$

from both molecules, including information on their neighborhoods, membership to certain structural elements and other characteristics. Then a valid graph kernel between  $m$ ,  $m'$  is the optimal assignment kernel [3]:

$$k_{OA}(m, m') := \begin{cases} \max_{\Pi} \sum_{h=1}^{|m|} k_{nei}(a_h, a'_{\Pi(h)}) & \text{if } |m'| \geq |m| \\ \max_{\Pi} \sum_{j=1}^{|m'|} k_{nei}(a_{\Pi(h)}, a'_h) & \text{otherwise} \end{cases} \quad (3)$$

where  $k_{nei}$  is calculated based on two kernels  $k_{atom}$  and  $k_{bond}$  which compare the atom and bond features, respectively. Each atom of the smaller of both molecules is assigned to exactly one atom of the larger molecule such that the overall similarity score is maximized. To prevent larger molecules automatically achieving a higher kernel value than smaller ones, kernel is normalized (Schoelkopf & Smola, 2002) [10], i.e.

$$k_{OA}(m, m') \leftarrow \frac{k_{OA}(m, m')}{\sqrt{k_{OA}(m, m)k_{OA}(m', m')}} \quad (4)$$

OA graph kernel, is of general use for QSAR problems, but is more focused on screening data. In developing a good quality QSAR model for these data we combined OA graph kernel and standard kernel (Gaussian) where we considered Burden descriptors [1].

$$k_{Gaus}(m, m') = \exp(-\gamma|m - m'|) \quad (5)$$

Calculating the OA graph kernel and at same time the Gaussian kernel with eight Burden features give us two ways of comparing molecules: looking at the chemical structure and looking at molecular activity. Having two values from these two kernels and making, at the same time an average, gives us more precise molecule similarity information.

$$k_{SCR}(m, m') = \frac{k_{Gaus}(m, m') + K_{OA}(m, m')}{2} \quad (6)$$

To improve accuracy of classification in screening data using introduced screening kernel we want also optimized parameters in Gaussian kernel where we use as properties Burden descriptors [1]. The parameter  $\gamma$  in Gaussian kernel can be determined by the *loo* bound described below. Vapnik [12] showed that the following radius margin bound holds for SVM without the bias term  $b$ .

$$loo \leq 4R^2\|w\|^2 \quad (7)$$

where *loo* is the number of leave-one-out error,  $w$  is the solution of (1) and  $R$  is the radius of the smallest sphere. The  $R^2$  can be obtained as the objective function value of the following optimization problem:

$$\max 1 - \beta^T K_{i,j} \beta \quad (8)$$

$$\text{subject to } \sum_{i=1}^l \beta_i = 1, \beta_i \geq 0, i = 1, 2, \dots, l, \text{ where } K_{ij} = K(m_i, m'_j) \quad (9)$$

The optimization problem [4] is needed to solve to get the radius in each iterative step for minimizing the *loo* bound. As we consider the Gaussian kernel as one part of screening kernel, we may think that data structure of the input data in the feature space changes by changing the kernel parameter. Intuitively, we get different distribution of the same input data in the feature space by changing the kernel parameter. From the error bound in equation (7), we see that  $R^2$  and  $\|w\|^2$  both depend on the structure of feature data. Therefore, the performance of the SVM depends on the distribution of the input data in the feature space. For our Gaussian kernel used with graph kernel we analyze the distribution of data in the feature space on the basis of the principal component analysis (PCA) method, and apply a simple method that predicts a good set of kernel parameters quickly.

### 3 PCA

Performing PCA in kernel space, we assume that  $p$  dimensional linear space is transformed to  $d$  dimensional space. Let  $n_i$  be an orthonormal vector of  $k$ -dimensional linear subspace, and let  $A = (n_1, n_2, \dots, n_k)$ . Then,  $\Phi(m)$  is transformed into  $H$  subspace of space  $F$  by the linear operator  $A$  as

$$\psi(m_i) = A^T \Phi(m_i) \in R^k. \tag{10}$$

Let  $1$  be a  $1 \times 1$  matrix of all ones and  $\Phi = ((\Phi(m_1), \Phi(m_2), \dots, \Phi(m_l)))$ . The covariance matrix of feature vectors  $\Phi$  in the feature space  $F$  is given by

$$\sum_{\Phi} = \frac{1}{l} \Phi (I - \frac{1}{l}) (I - \frac{1}{l}) \Phi^T \tag{11}$$

The covariance matrix of  $\Phi(m_i)$  in the  $k$ -dimensional subspace is given by

$$\sum_{\psi} = A^T \sum_{\Phi} A. \tag{12}$$

The orthonormal bases  $n_i$  lie in the span of  $(\Phi(m_1), \Phi(m_2), \dots, \Phi(m_l))$ . We can represent them by the linear combination of  $(\Phi(m_i) - \Phi)$  inf where  $\Phi = \sum_{i=1}^l \Phi(m_i)$  (centering the data). Then,  $n_i$  can be written as

$$n_i = \sum_{j=1}^l s_{ij} (\Phi(m_j) - \Phi), \tag{13}$$

where  $s_i = (s_{i1}, s_{i2}, \dots, s_{ik})^T$  is a coefficient vector.

Letting  $S = (s_1, s_2, \dots, s_k)$ ,

$$A = \Phi (I - \frac{1}{l}) S. \tag{14}$$

Let  $G = (I - \frac{1}{l}) \Phi^T \Phi (I - \frac{1}{l})$  where the kernel matrix is  $K = \Phi^T \Phi$ . The covariance matrix of feature vectors is given by

$$\sum_{\psi} = \frac{1}{l} S^T G^2 S, \tag{15}$$

and

$$A^T A = G^T G S \quad (16)$$

The  $S$  can be calculated according to the PCA variance maximization criterion [6][7], i.e., maximizing variances such that  $A^T A = I$ . This gives the Lagrangian  $L(S; \Lambda) = (\frac{1}{l})tr(S^T G^2 S) - tr((S^T G^2 S - I)\Lambda)$  with Lagrange multiplier  $\Lambda > 0$  where is a diagonal matrix. Then setting

$$\frac{\theta L}{\theta S} = \frac{2}{l}G^2 S - 2G S \Lambda \quad (17)$$

by zero, we obtain Karush-Kuhn-Tucker(KKT) condition

$$\frac{1}{l}G^2 S = G S \Lambda, \quad S^T G S = I_k \quad (18)$$

Let,  $i$ -th eigenvalue of  $G$  be  $\mu_i$  with corresponding eigenvectors  $v_i$ , and  $\mu_1, \mu_2, \dots, \mu_k \geq 0$ . Since  $G$  is symmetric matrix, eigenvectors constitute orthonormal set. We denote  $V = (v_1, v_2, \dots, v_k)$ , and a  $k \times k$  diagonal matrix  $M$  whose  $M_{ii} = \mu_i$ . If we consider  $S = V M^{-\frac{1}{2}}$  where

$$\left[ M^{-\frac{1}{2}} \right]_{i,j} = \begin{cases} \frac{1}{\sqrt{\mu_i}} & \text{if } i=j, \\ 0 & \text{if } i \neq j \end{cases} \quad (19)$$

And eq. (18) can be transformed to

$$(left\ side) = \frac{1}{l}G^2 S = \frac{1}{l}V M^2 M^{-\frac{1}{2}} \text{ and } (right\ side) = G S \Lambda = V M M^{-\frac{1}{2}} \Lambda \quad (20)$$

by using  $GV = VM$  and  $V, M$  are non zero matrix.

Since  $GV = VM$ ,

$$\begin{aligned} \sum_{\psi} &= \frac{1}{l}S^T G^2 S = \frac{1}{l}M^{-\frac{1}{2}} V^T G^2 V M^{-\frac{1}{2}} = \frac{1}{l}M^{-\frac{1}{2}} V^T G V M M^{-\frac{1}{2}} = \\ &= \frac{1}{l}M^{-\frac{1}{2}} V^T V M M^{\frac{1}{2}} = \frac{1}{l}M^{-\frac{1}{2}} I M M^{\frac{1}{2}} = \frac{1}{l}M. \end{aligned} \quad (21)$$

Thus eigenvalues of  $\sum_{\psi}$  equal to the eigenvalues of  $(1/l)G$ . From the eigenvalues and corresponding eigenvectors we may get the abstraction of the data. Usually, eigenvalues determine the structure of the data. This is immaterial since we need only the positive eigenvalues of  $G$ .

## 4 Kernel Parameter Optimization

For optimization of  $\gamma$  parameter we used method presented by Debnath and Takahashi [7]. Eigenvalues,  $\mu$ 's, are large for some kernel parameters  $\gamma$  means data spread in a large area. So we may not get good performance for those  $\gamma$  for which eigenvalues are large enough. If eigenvalues,  $\mu_i$ 's are too small, however the

number of positive eigenvalues are large, data lie in a very compact area. Thus the number of support vectors will be large and the support vector machine may implement a rapidly oscillating function rather than the smooth mapping. As a results, the performance will not be good. In [4], Chung et. al. showed that a good bound should avoid minima that happens at the boundary (i.e., too small and too large  $\gamma$  do not show good performance). In [8], Keerthi and Lin also showed that too small creates overfitting/underfitting problems, and too large  $\gamma$  creates underfitting problem. For large and small value of kernel parameter  $\gamma$ , our ideas follow Chung et. al. and Keerthi and Lin’s idea. When the number of positive eigenvalues of  $G$  is small enough, many feature data linearly depend on other feature data. Let the  $k$ -th eigenvalue of  $G$  be  $\mu_k$  with corresponding eigenvectors  $v_k \in R_l$  and  $\Phi'(x_i) = \Phi(x_i) - \Phi$ .

$$\begin{aligned} \mu_k &= v_K^T G v_k = v_K^T (I - \frac{1}{\gamma}) v_K = (\sum_{i=1}^l v_{ki} (\Phi(x_i) - \Phi)^T \sum_{j=1}^l v_{kj} (\Phi(x_j) - \Phi)^T) = \\ &= (\sum_{i=1}^l v_{ki} \Phi'(x_i))^T \sum_{j=1}^l v_{kj} \Phi'(x_j) \end{aligned} \quad (22)$$

As the  $K = \Phi^T \Phi$  is the rank one matrix when  $\gamma \rightarrow \infty$ , many eigenvalues of  $G$  become zero (or smaller than a small threshold value) when the value of  $\gamma$  is large. The  $\mu_k$  is zero means  $\sum_{i=1}^l v_{ki} \Phi'(x_i) = 0$ , i.e. some data are linearly depend on other data. So for large value of  $\gamma$ , we loose information for many data as some feature data are becoming the linear combination of others, and the kernel matrix  $K$  does not have their information. The small number of positive eigenvalues means the SVM learns with a small fraction of training data from the training set. Thus, very small number of positive eigenvalues can not show good performance. Considering all above conditions, good kernel parameters must avoid either too large or too small eigenvalues, and the  $G$  has sufficient number of positive eigenvalues. Therefore, given a set of kernel parameters we choose a subset of kernel parameters for which  $G$  has sufficient number of positive eigenvalues and the largest eigenvalue  $\mu_1$ , is not too large.

## 5 Experimental Evaluations and Results

We experimentally evaluated the performance of graphical kernel where parameters are optimized in a classification algorithm and compared it against that achieved by earlier kernels on a variety of chemical compound datasets from the National Cancer Institute’s DTP AIDS Anti-viral Screen program [9] and from the Center of Computational Drug Discovery’s anthrax project at the University of Oxford [5]. For these datasets we generated 8 features, called Burden descriptors [1] (eight dimensional spaces).

We tested a proposed extension of molecule graph kernel in two benchmark experiments of chemical compound classification. We use the standard SVM algorithm for binary classification described previously. The regularization factor

**Table 1.** Recognition performance comparison of  $K_{OA}$  with  $K_{scr}$  kernel and  $K_{scr}$  with optimized parameters ( $K_{scrOP}$ ) for different number of training sets. Numbers represent correct classification rate [%]

Training set	OA Kernel			$K_{SCR}$			$K_{scrOP}$		
	DTP	Toxic	Diff.	DTP	Toxic	Diff.	DTP	Toxic	Diff.
100	79.6	73.9	5.5	79.1	74.9	4.9	79.5	73.2	5.3
200	79.9	74.1	5.8	80.1	75.6	4.5	81.0	77.1	3.9
300	80.1	79.1	1.0	80.8	79.6	1.2	82.2	80.3	1.9
400	81.3	79.8	1.5	82.5	80.3	2.2	81.9	80.8	1.1
500	82.5	81.3	1.2	83.2	81.5	1.8	83.6	82.0	1.6

of SVM was fixed to  $C = 10$ . In order to see the effect of generalization performance on the size of training data set and model complexity, experiments were carried out by varying the number of training samples (100, 200, 300, 400, 500) according to a 5-fold cross validation evaluation of the generalization error. The  $K_{scrOP}$  shows performance improvements over the standard  $K_{scr}$  and  $K_{OA}$  kernel, for both of the (noisy) and real data sets. Moreover it is worth mentioning that  $K_{scr}$  does slightly better than  $K_{OA}$  kernel in general. We would like now to determine if optimized  $K_{scrOP}$  kernel has an effect in SVM classification in comparison to  $K_{scr}$  and  $K_{OA}$ . A permutation test was selected as an alternative way to test for differences in our kernels in a nonparametric fashion. The permutation test based on algorithm [13] 2000 sample replacements estimated a  $p$ -value to decide whether or not to reject the null hypothesis. The null hypotheses for this test were  $H01 : K_{scrOP} = K_{scr}$ , and alternative hypothesis  $HA1 : K_{scrOP} > K_{scr}$ , additionally let's assume at a significance level  $\alpha = 0.05$ . The permutation test will reject the null hypothesis if the estimated  $P$ -value is less than  $\alpha$ . More specifically, for any value of  $\alpha < p - value$ , fail to reject  $H0$ , and for any value of  $\alpha = P - value$ , reject  $H0$ . The  $P$ -values on average for DTS AIDS and toxic data sets of 0.0527, 0.0025, 0.0426, 0.0020 indicates that the SVM with  $K_{scrOP}$  kernel is probably not equal to  $K_{scr}$  kernel. The  $P$ -value 0.0527 between  $K_{OP}$  and  $K_{scr}$  kernel for DTS AIDS data sets, indicates weak evidence against the null hypothesis. There is strong evidence that all other tests null hypothesis can be rejected. Permutation tests suggest, on average, that  $K_{scrOA}$  kernel for screening data is statistically significantly larger than  $K_{OA}$  and RBF.

## 6 Conclusion

We propose a method for finding good kernel parameter in graph kernel for chemical compounds only applying the eigen analysis method that is suitable for discriminative classification with unordered sets of local features. We have applied tuned gaussian kernel in our screening kernel to SVM-based object recognition tasks, and demonstrated recognition performance with accuracy comparable to existing kernels on screening data. Our experimental evaluation showed

that parameter tuning leads to substantially better results than those obtained by existing same kernel without optimization.

## References

1. Burden, F.R. 1989. "Molecular Identification Number For Substructure Searches", *Journal of Chemical Information and Computer Sciences*, 29, 225-227.
2. Graham W. Richards. Virtual screening using grid computing: the screensaver project. *Nature Reviews: Drug Discovery*, 1:551-554, July 2002.
3. H. Froehlich, J. K. Wegner, A. Zell, *QSAR Comb. Sci.* 2004, 23, 311 - 318.
4. K.-M. Chung, W.-C. Kao, T. Sun, L.-L. Wang, and C.-J. Lin, "Radius margin bounds for support vector machines with Gaussian kernel," *Neural Computation*, vol. 15, pp. 2643-2681, 2003.
5. Kozak K., Kozak M., Stapor K., *Kernels for chemical compounds in biological screening. ICANNGA 2007*, Springer-Verlag, LNCS/LNAI Proceedings
6. Pearlman, R. S. and Smith, K. M. 1998. "Novel software tools for chemical diversity", *Perspectives in Drug Discovery and Design*, 9/10/11, 339-353.
7. R. Debnath, H. Takahashi, "Generalization of kernel PCA and automatic parameter tuning," *Neural Information Processing*, Vo. 5, No. 3, December 2004.
8. S. S. Keerthi and C.-J. Lin, "Asymptotic behaviours of support vector machines with Gaussian kernel," *Neural Computation*, vol. 15, pp. 1667-1689, 2003.
9. S. Kramer, L. De Raedt, and C. Helma. *Molecular feature mining in hiv data*. In 7th Inter-national Conference on Knowledge Discovery and Data Mining, 2001.
10. Schoelkopf, B., and Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA, MIT Press.
11. V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
12. V. Vapnik and O. Chapelle, "Bounds on error expectation for support vector machines," *Neural computation*, vol. 12, no. 9, pp. 2013-2036, 2000.
13. Westfall, P. H. and Young, S. S. (1993). *Resampling-based multiple testing: Examples and methods for p-value adjustment*, John Wiley and Sons.

---

# Estimation of the Preference Relation on the Basis of Medians from Pairwise Comparisons in the Form of Difference of Ranks

Leszek Klukowski

Systems Research Institute Polish Academy of Sciences, 6 Newelska Str., 01-447  
Warsaw, Poland  
Leszek.Klukowski@ibspan.waw.pl

**Summary.** The problem of estimation of the preference relation in a finite set on the basis of medians from multiple pairwise comparisons, in the form of difference of ranks with random errors, with the use of nearest adjoining order idea (NAO), is investigated in the paper. The estimated form of the relation is obtained on the basis of a discrete programming problem; some probabilistic properties of the estimator are determined. They are obtained under weak assumptions about distributions of comparison errors; in particular, the expected value of comparison error of each pair can be different than zero.

## 1 Introduction

The paper presents an extension of the method of ranking of elements, from a finite set, on the basis of pairwise comparisons, presented in Klukowski [3], [4]. The results discussed in Klukowski [3] relate to the case of (multiple) comparisons indicating the direction of preference, while the paper Klukowski [4] examines the case of one comparison of each pair in the form of difference of ranks. The extension presents the approach based on medians from multiple comparisons, which is similar to the case of majority principle examined in Klukowski [3], section 5.2. It exploits the idea of nearest adjoining order (NAO) introduced by Slater [7] (see also David [2]). The results presented in the paper are based on weak assumptions about distributions of comparison errors; it is sufficient, that each distribution is unimodal with maximum and median in zero (the probability functions of comparison errors may be unknown). The median approach is simple from computational point of view, especially it allows to use the same optimisation algorithm, as in the case of one comparison of each pair. The empirical results, based on simulation approach, confirm the usefulness of the method.

The paper consists of five sections. Main results: the problem formulation, definitions and notations, the form of the estimator and its properties are presented in Sections 2 – 4. Last section comprises summary.

## 2 Problem Formulation

The formulation of the problem is an extension of the problem stated in Klukowski [4] for the case of  $N > 1$  independent comparisons of each pair.



It is assumed, that in a finite set of elements  $\mathbf{X} = \{x_1, \dots, x_m\}$ ,  $m \geq 3$ , there exists an unknown complete weak preference relation  $\mathbf{R}$  of the form:

$$\mathbf{R} = \mathbf{I} \cup \mathbf{P}, \tag{1}$$

where:

- $\mathbf{I}$  - the equivalence relation (reflexive, transitive, symmetric),
- $\mathbf{P}$  - the strict preference relation (transitive, asymmetric).

The preference relation  $\mathbf{R}$  generates from elements of the set  $\mathbf{X}$  the sequence of subsets  $\chi_1^*, \dots, \chi_n^*$  ( $n \leq m$ ), such that each element  $x_i \in \chi_r^*$  is preferred to any element  $x_j \in \chi_s^*$  ( $r < s$ ) and each subset  $\chi_q^*$  ( $1 \leq q \leq n$ ) comprises equivalent elements only.

The relation  $\mathbf{R}$  can be characterized by the function  $T : \mathbf{X} \times \mathbf{X} \rightarrow D$ ,  $D = \{-(n - 1), \dots, 0, \dots, n - 1\}$ , defined as follows:

$$T(x_i, x_j) = d_{ij} \Leftrightarrow x_i \in \chi_k^*, \quad x_j \in \chi_l^*, \quad d_{ij} = k - l. \tag{2}$$

The value of the function  $T(x_i, x_j)$  is the difference of ranks of the elements  $x_i$  and  $x_j$  in the relation  $\mathbf{R}$ . In the case when  $T(x_i, x_j) < 0$ , ( $T(x_i, x_j) > 0$ ) the element  $x_i$  precedes element  $x_j$  (element  $x_j$  precedes  $x_i$ ), for  $|d_{ij}|$  positions. The value  $T(x_i, x_j) = 0$  means the equivalence of both elements (they belong to the same subset  $\chi_q^*$ ,  $1 \leq q \leq n$ ). It is obvious, that  $T(x_i, x_j) = -T(x_j, x_i)$  for  $T(\cdot) \neq 0$ .

The relation form is to be determined (estimated) on the basis of pairwise comparisons of elements of the set  $\mathbf{X}$ , disturbed with random errors. Each pair  $(x_i, x_j) \in \mathbf{X}$  is compared independently (in stochastic sense)  $N$  times; the result of  $k$ -th comparison ( $k = 1, \dots, N$ ;  $N > 1$ ) is the value of the function:

$$g_k : \mathbf{X} \times \mathbf{X} \rightarrow D, \quad D = \{-(m - 1), \dots, m - 1\}; \tag{3}$$

the result  $g_k(x_i, x_j) = c_{kij}$  is an assessment of the difference of ranks of the pair  $(x_i, x_j)$ , in  $k$ -th comparison. The set  $D$  can include values from the range:  $-(m - 1), \dots, m - 1$  because the number of subsets  $n$  is assumed unknown.

In the paper it is assumed, that each comparison  $g_k(x_i, x_j)$  ( $1 \leq k \leq N$ ) can be disturbed by random error; it means, that the difference  $T(x_i, x_j) - g_k(x_i, x_j)$  may assume values different than zero - with some probabilities. The comparisons  $g_k(x_i, x_j)$  and  $g_l(x_r, x_s)$  are independent, i.e.:

$$\begin{aligned} &Pr((g_k(x_i, x_j) = c_{kij}) \cap (g_l(x_r, x_s) = c_{lrs})) = \\ &Pr(g_k(x_i, x_j) = c_{kij})Pr(g_l(x_r, x_s) = c_{lrs}) (k \neq l). \end{aligned} \tag{4}$$

The probabilities, which characterize each distribution of comparison errors will be denoted with the symbols  $\alpha_{ijk}(l)$ ,  $\beta_{ijk}(l)$ ,  $\gamma_{ijk}(l)$ ; the probabilities are defined as follows:

$$\begin{aligned} \alpha_{ijk}(l) = &Pr(T(x_i, x_j) - g_k(x_i, x_j) = l; T(x_i, x_j) = 0) \\ &(- (m - 1) \leq l \leq m - 1), \end{aligned} \tag{5}$$

$$\beta_{ijk}(l) = \Pr(T(x_i, x_j) - g_k(x_i, x_j) = l; T(x_i, x_j) < 0) - 2(m - 1) \leq l \leq 2(m - 1)), \tag{6}$$

$$\gamma_{ijk}(l) = \Pr(T(x_i, x_j) - g_k(x_i, x_j) = l; T(x_i, x_j) > 0) (-2(m - 1) \leq l \leq 2(m - 1)). \tag{7}$$

It is obvious, that the probabilities (5)-(7) have to fulfill the equalities:

$$\sum_{l=-(m-1)}^{m-1} \alpha_{ijk}(l) = 1, \quad \sum_{l=-2(m-1)}^{2(m-1)} \beta_{ijk}(l) = 1, \quad \sum_{l=-2(m-1)}^{2(m-1)} \gamma_{ijk}(l) = 1. \tag{8}$$

Moreover it is assumed that the following assumptions hold:

$$\sum_{l \leq 0} \Pr(T(x_i, x_j) - g_k(x_i, x_j) = l) > 1/2, \tag{9}$$

$$\sum_{l \geq 0} \Pr(T(x_i, x_j) - g_k(x_i, x_j) = l) > 1/2, \tag{10}$$

$$\Pr(T(x_i, x_j) - g_k(x_i, x_j) = l) \geq \Pr(T(x_i, x_j) - g_k(x_i, x_j) = l + 1) \quad (l \geq 0), \tag{11}$$

$$\Pr(T(x_i, x_j) - g_k(x_i, x_j) = l) \geq \Pr(T(x_i, x_j) - g_k(x_i, x_j) = l - 1) \quad (l \leq 0). \tag{12}$$

The conditions (9)-(12) guarantee, that: • zero is the median of each distribution, • each probability function is unimodal and • assumes maximum in zero. The expected value of any comparison error can differ from zero.

The probabilities  $\alpha_{ijk}(0)$ ,  $\beta_{ijk}(0)$  and  $\gamma_{ijk}(0)$  may be lower than  $\frac{1}{2}$ ; therefore the assumptions about distributions of comparison errors are more general, than those in “direction of preference” approach (see Klukowski [3]).

For simplification it is assumed, that distribution of any comparison error  $T(x_i, x_j) - g_k(x_i, x_j) ((x_i, x_j) \in \mathbf{X} \times \mathbf{X})$  is the same for each  $k$ ,  $1 \leq k \leq N$  (as a result the comparisons of each pair  $g_1(x_i, x_j), \dots, g_N(x_i, x_j)$  are iid. random variables). Therefore, the index  $k$  will be omitted in the symbols:  $\alpha_{ij}(l)$ ,  $\beta_{ij}(l)$ ,  $\gamma_{ij}(l)$ . The relaxation of the assumption is not difficult.

The probabilities  $\alpha_{ij}(l) (-m - 1 \leq l \leq m - 1)$  determine the probability function of comparison errors for equivalent elements  $x_i$  and  $x_j$  (because  $T(x_i, x_j) = 0$ ). Each probability  $\alpha_{ij}(l)$  means, that an error of comparison equals  $l$ ; especially  $\alpha_{ij}(0)$  denotes the probability of errorless comparison (because  $T(x_i, x_j) = g_k(x_i, x_j) = 0$ ). In the case of known number of subsets  $n$  the interval of integers  $[-(m - 1), m - 1]$  ought to be replaced with the interval  $[-(n - 1), n - 1]$ . The interpretation of the probabilities  $\beta_{ij}(l)$  and  $\gamma_{ij}(l) (-2(m - 1) \leq l \leq 2(m - 1))$  is similar, with the difference, that they both determine distribution errors for non-equivalent elements.

The problem of the preference relation estimation with the use of the above definitions can be stated as follows.

To determine the relation  $\mathbf{R}$  (i.e. the sequence of subsets  $\chi_1^*, \dots, \chi_n^*$  or the values  $T(x_i, x_j)$ ) on the basis of the comparisons  $g_k(x_i, x_j) (k = 1, \dots, N)$ , made for each pair  $(x_i, x_j) \in \mathbf{X} \times \mathbf{X}$ .

### 3 Basic Definition and Notation

The following notation is necessary for further considerations:

- $t(x_i, x_j)$  - the function which determines the difference of ranks for each pair  $(x_i, x_j) \in \mathbf{X} \times \mathbf{X}$ , for any preference relation  $\chi_1, \dots, \chi_r$  in the set  $\mathbf{X}$ , i.e.:

$$t(x_i, x_j) = d_{ij} \Leftrightarrow x_i \in \chi_k, \quad x_i \in \chi_l; \quad d_{ij} = k - l, \quad (13)$$

- $I(\chi_1, \dots, \chi_r), P_1(\chi_1, \dots, \chi_r), P_2(\chi_1, \dots, \chi_r)$  - the sets of pairs of indices  $\langle i, j \rangle, (j > i)$ , generating the relation  $\chi_1, \dots, \chi_r$ , i.e.:

$$I(\chi_1, \dots, \chi_r) = \{ \langle i, j \rangle \mid t(x_i, x_j) = 0; j > i \}, \quad (14)$$

$$P_1(\chi_1, \dots, \chi_r) = \{ \langle i, j \rangle \mid t(x_i, x_j) < 0; j > i \}, \quad (15)$$

$$P_2(\chi_1, \dots, \chi_r) = \{ \langle i, j \rangle \mid t(x_i, x_j) > 0; j > i \}; \quad (16)$$

- $R_m = I(\chi_1, \dots, \chi_r) \cup P_1(\chi_1, \dots, \chi_r) \cup P_2(\chi_1, \dots, \chi_r) =$

$$\{ \langle i, j \rangle \mid 1 \leq i, j \leq m; j > i \}. \quad (17)$$

The basis for the properties of the estimator of the preference relation examined in the paper are the variables  $U_{ij}^{(k)}(\chi_1, \dots, \chi_r), V_{ij}^{(k)}(\chi_1, \dots, \chi_r), Z_{ij}^{(k)}(\chi_1, \dots, \chi_r), W^{(k)}(\chi_1, \dots, \chi_r)$  defined, as follows:

$$U_{ij}^{(k)}(\cdot) = |g_k(x_i, x_j)|; \quad t(x_i, x_j) = 0, \quad (18)$$

$$V_{ij}^{(k)}(\cdot) = |t(x_i, x_j) - g_k(x_i, x_j)|; \quad t(x_i, x_j) < 0, \quad (19)$$

$$Z_{ij}^{(k)}(\cdot) = |t(x_i, x_j) - g_k(x_i, x_j)|; \quad t(x_i, x_j) > 0, \quad (20)$$

$$W^{(k)}(\cdot) = \sum_{\langle i, j \rangle \in I(\cdot)} U_{ij}^{(k)}(\cdot) + \sum_{\langle i, j \rangle \in P_1(\cdot)} V_{ij}^{(k)}(\cdot) + \sum_{\langle i, j \rangle \in P_2(\cdot)} Z_{ij}^{(k)}(\cdot). \quad (21)$$

The random variables and other symbols corresponding to the relation  $\mathbf{R}$  (errorless result of the estimation problem) will be marked with asterisks, i.e.:  $U_{ij}^{(k)*}, V_{ij}^{(k)*}, Z_{ij}^{(k)*}, I^*, P_1^*, P_2^*, W^{(k)*}$ , while variables and symbols corresponding to any other relation  $\tilde{\chi}_1, \dots, \tilde{\chi}_{\tilde{n}}$ , different than errorless one, will be denoted:  $\tilde{U}_{ij}^{(k)}, \tilde{V}_{ij}^{(k)}, \tilde{Z}_{ij}^{(k)}, \tilde{I}, \tilde{P}_1, \tilde{P}_2, \tilde{W}^{(k)}$ . For fixed  $k$  ( $1 \leq k \leq N$ ) the difference  $W^{(k)*} - \tilde{W}^{(k)}$  can be written in the form:

$$\begin{aligned} W^{(k)*} - \tilde{W}^{(k)} &= \sum_{I^* \cap (\tilde{P}_1 - P_1^*)} (U_{ij}^{(k)*} - \tilde{V}_{ij}^{(k)}) + \sum_{I^* \cap (\tilde{P}_2 - P_2^*)} (U_{ij}^{(k)*} - \tilde{Z}_{ij}^{(k)}) + \\ &+ \sum_{P_1^* \cap (\tilde{I} - I^*)} (V_{ij}^{(k)*} - \tilde{U}_{ij}^{(k)}) + \sum_{P_1^* \cap (\tilde{P}_2 - P_2^*)} (V_{ij}^{(k)*} - \tilde{Z}_{ij}^{(k)}) + \\ &+ \sum_{(P_1^* \cap \tilde{P}_1) \cap (T(\cdot) \neq \tilde{I}(\cdot))} (V_{ij}^{(k)*} - \tilde{V}_{ij}^{(k)}) + \sum_{P_2^* \cap (\tilde{I} - I^*)} (Z_{ij}^{(k)*} - \tilde{U}_{ij}^{(k)}) + \\ &+ \sum_{P_2^* \cap (\tilde{P}_1 - P_1^*)} (Z_{ij}^{(k)*} - \tilde{V}_{ij}^{(k)}) + \sum_{(P_2^* \cap \tilde{P}_2) \cap (T(\cdot) \neq \tilde{I}(\cdot))} (Z_{ij}^{(k)*} - \tilde{Z}_{ij}^{(k)}) \end{aligned} \quad (22)$$

or shortly in the form:

$$W^{(k)*} - \tilde{W}^{(k)} = \sum_{\nu=1}^8 \sum_{S_\nu} Q_{ij}^{(k\nu)},$$

where:

$$Q_{ij}^{(k,1)} = U_{ij}^{(k)*} - \tilde{V}_{ij}^{(k)}, \quad S_1 = \{ \langle i, j \rangle \mid \langle i, j \rangle \in I^* \cap (\tilde{P}_1 - P_1^*) \}, \quad (23)$$

$$Q_{ij}^{(k,2)} = U_{ij}^{(k)*} - \tilde{Z}_{ij}^{(k)}, \quad S_2 = \{ \langle i, j \rangle \mid \langle i, j \rangle \in I^* \cap (\tilde{P}_2 - P_2^*) \}, \quad (24)$$

$$Q_{ij}^{(k,3)} = V_{ij}^{(k)*} - \tilde{U}_{ij}^{(k)}, \quad S_3 = \{ \langle i, j \rangle \mid \langle i, j \rangle \in P_1^* \cap (\tilde{I} - I^*) \}, \quad (25)$$

$$Q_{ij}^{(k,4)} = V_{ij}^{(k)*} - \tilde{Z}_{ij}^{(k)}, \quad S_4 = \{ \langle i, j \rangle \mid \langle i, j \rangle \in P_1^* \cap (\tilde{P}_2 - P_2^*) \}, \quad (26)$$

$$Q_{ij}^{(k,5)} = V_{ij}^{(k)*} - \tilde{V}_{ij}^{(k)}, \quad S_5 = \{ \langle i, j \rangle \mid \langle i, j \rangle \in (P_1^* \cap \tilde{P}_1) \cap (T(x_i, x_j) \neq \tilde{t}(x_i, x_j)) \}, \quad (27)$$

$$Q_{ij}^{(k,6)} = Z_{ij}^{(k)*} - \tilde{U}_{ij}^{(k)}, \quad S_6 = \{ \langle i, j \rangle \mid \langle i, j \rangle \in P_2^* \cap (\tilde{I} - I^*) \}, \quad (28)$$

$$Q_{ij}^{(k,7)} = Z_{ij}^{(k)*} - \tilde{V}_{ij}^{(k)}, \quad S_7 = \{ \langle i, j \rangle \mid \langle i, j \rangle \in P_2^* \cap (\tilde{P}_1 - P_1^*) \}, \quad (29)$$

$$Q_{ij}^{(k,8)} = Z_{ij}^{(k)*} - \tilde{Z}_{ij}^{(k)}, \quad S_8 = \{ \langle i, j \rangle \mid \langle i, j \rangle \in P_2^* \cap \tilde{P}_2 \cap (T(x_i, x_j) \neq \tilde{t}(x_i, x_j)) \}. \quad (30)$$

The following properties of the random variables  $Q_{ij}^{(k,\nu)}$  are indispensable for further considerations.

**Lemma 1.** *The expected value of each random variable  $Q_{ij}^{(k,\nu)}$  ( $1 \leq k \leq N$ ;  $\langle i, j \rangle \in S_\nu$ ;  $\nu = 1, \dots, 8$ ) satisfy the condition:*

$$E(Q_{ij}^{(k,\nu)}) < 0. \quad (31)$$

Proof - see Klukowski [6].

## 4 The Estimator of the Relation Based on Medians from Comparisons

In the case of median of comparisons it is assumed, that  $N$  is odd, i.e.  $N = 2\tau + 1$  ( $\tau = 0, 1, \dots$ ). The basis for the estimation of the relation form are the random variables:  $U_{me,N}(\cdot)$ ,  $V_{me,N}(\cdot)$ ,  $Z_{me,N}(\cdot)$ ,  $W_{me,N}(\cdot)$  defined in the following way:

$$U_{me,N}(x_i, x_j) = |g_{me,N}(x_i, x_j)|; \quad t(x_i, x_j) = 0, \quad (32)$$

$$V_{me,N}(x_i, x_j) = |t(x_i, x_j) - g_{me,N}(x_i, x_j)|; \quad t(x_i, x_j) < 0, \quad (33)$$

$$Z_{me,N}(x_i, x_j) = |t(x_i, x_j) - g_{me,N}(x_i, x_j)|; \quad t(x_i, x_j) > 0, \quad (34)$$



$$Pr(W_{me,N}^* < \tilde{W}_{me,N}) \geq - \frac{E(\sum_{\nu=1}^8 \sum_{S_\nu} Q_{ij,me,N}^{(\nu)})}{\lambda_1(m-1) + 2\lambda_2(m-1) + \lambda_3(m-2)}, \tag{39}$$

where:

$\lambda_1 = \#(S_1 \cup S_2 \cup S_3 \cup S_6)$ ;  $\lambda_2 = \#(S_4 \cup S_7)$ ;  $\lambda_3 = \#(S_5 \cup S_8)$ ;  $\#(\Xi)$  – number of elements of the set  $\Xi$ .

Proof of the inequalities (38), (39) – see appendix.

The inequalities (38), (39) indicate that: the random variable  $W_{me,N}^*$ , corresponding to the relation  $\chi_1^*, \dots, \chi_n^*$ , assumes minimal expected value in the set  $W_{me,N}(\chi_1, \dots, \chi_r)$  and the probability  $Pr(W_{me,N}^* < \tilde{W}_{me,N})$  is not lower than the right-hand side of the inequality (39). Therefore, it is rational to determine the relation  $\hat{\chi}_1, \dots, \hat{\chi}_n$ , which minimizes the value of the variable  $W_{me,N}(\chi_1, \dots, \chi_r)$  for given comparisons  $g_{me,N}(x_i, x_j)$  ( $(x_i, x_j) \in \mathbf{X} \times \mathbf{X}$ ), and to assume it as the estimate of the relation. The optimisation task for estimation problem assumes the form:

$$\min_{\chi_1^{(\iota)}, \dots, \chi_{r^{(\iota)}}^{(\iota)} \in F_X} \left\{ \sum_{\langle i,j \rangle \in R_m} |t^{(\iota)}(x_i, x_j) - g_{me,N}(x_i, x_j)| \right\}, \tag{40}$$

where:

- $F_X$  – the feasible set – a family of all preference relations in the set  $\mathbf{X}$ ;
- $\chi_1^{(\iota)}, \dots, \chi_{r^{(\iota)}}^{(\iota)}$  – an element of the set  $F_X$ ;
- $t^{(\iota)}(x_i, x_j)$  – the function characterizing the relation  $\chi_1^{(\iota)}, \dots, \chi_{r^{(\iota)}}^{(\iota)}$ .

The number of solutions of the task (40) may exceed one. The unique solution can be selected randomly or with the use of an additional criterion.

The right-hand side of the inequality (39) is included in the interval (0, 1). Its numerical value can be determined in the case of known distributions of comparison errors  $Pr(T(\cdot) - g_k(\cdot) = l)$ . In opposite case it can be approximated. However, the method proposed in the paper can be used in the case of unknown distributions of comparison errors; it is sufficient to satisfy the conditions (9)- (12).

It should be emphasized, that the evaluation (39) is based on rough probability inequality and does not indicate exponential convergence to one, as in the case of comparisons indicating direction of preference (see Klukowski [3], point 5.2). However, the differences of ranks can be transformed into comparisons expressing “direction of the preference”, which satisfy the condition, that probability of errorless comparison is higher than  $\frac{1}{2}$  (required in the paper mentioned). The idea of the transformation results from the formulas (A3) and (A4) presented in Appendix. It seems possible to prove that properties of the approach based on comparisons in the form of (multiple) differences of ranks are not worse than the approach based on transformed variables.

## 5 Summary

The paper presents the method of estimation of the preference relation on the basis of multiple pairwise comparisons in the form of difference of ranks with random errors. It is based on weak assumptions about distributions of comparison errors; the estimated form of the relation is obtained on the basis of the discrete optimisation problem. It seems necessary to investigate the efficiency of the estimator proposed with the use of simulation approach.

## References

1. David H. A. (1970) Order Statistics. J. Wiley, New York.
2. David H. A. (1988) The Method of Paired Comparisons, 2<sup>nd</sup> ed. Ch. Griffin, London.
3. Klukowski L. (1994) Some probabilistic properties of the nearest adjoining order method and its extensions. Annals of Operations Research, **51**, 241-261.
4. Klukowski L. (2000) The nearest adjoining order method for pairwise comparisons in the form of difference of ranks. Annals of Operations Research, **97**, 357 -378.
5. Klukowski L. (2007a) Estimation of tolerance relation the basis of multiple pairwise comparisons with random errors. Control and Cybernetics, to appear.
6. Klukowski L. (2007b) Estimation of preference relation the basis of multiple pairwise comparisons with random errors. Modeling of Preference and Risk'07 Conference in Ustronie, 16 - 17 April, 2007, Economics University, Katowice, to appear (in Polish).
7. Slater P. (1961) Inconsistencies in a schedule of paired comparisons. Biometrika, **48**, 303–312.

## A Appendix

*Proof of the inequality (38).*

The inequality (38) is true for  $N=1$  (on the basis of the inequality (31) – see Lemma 1 in Section 3). For  $N = 2\tau + 1$  ( $\tau = 1, \dots$ ) the proof is similar to the case considered in Klukowski [5]; therefore its draft is presented only. The probability function  $P(T(x_i, x_j) - g_{me,N}(x_i, x_j) = l)$  ( $N = 2\tau + 1; \tau = 0, 1, \dots$ ) satisfies, for each pair  $(x_i, x_j) \in \mathbf{X} \times \mathbf{X}$ , the inequalities:

$$Pr(T(x_i, x_j) - g_{me,N+2}(x_i, x_j) = 0) > Pr(T(x_i, x_j) - g_{me,N}(x_i, x_j) = 0); \quad (A1)$$

$$Pr(T(x_i, x_j) - g_{me,N+2}(x_i, x_j) = l) < Pr(T(x_i, x_j) - g_{me,N}(x_i, x_j) = l) \\ (l \neq 0). \quad (A2)$$

The inequalities (A1) and (A2) result from the following facts. The probabilities:  $Pr(T(x_i, x_j) - g_{me,N}(x_i, x_j) = l)$  can be expressed in the form (see David [1], section 2.4):

$$\begin{aligned}
 &Pr(T(x_i, x_j) - g_{me,N}(x_i, x_j) = 0) = \\
 &= Pr(T(x_i, x_j) - g_{me,N}(x_i, x_j) \leq 0) - Pr(T(x_i, x_j) - g_{me,N}(x_i, x_j) \leq -1) = \\
 &= \frac{N!}{(((N - 1)/2)!)^2} \int_{G(-1)}^{G(0)} t^{(N-1)/2}(1 - t)^{(N-1)/2} dt, \tag{A3}
 \end{aligned}$$

$$\begin{aligned}
 &Pr(T(x_i, x_j) - g_{me,N}(x_i, x_j) = l) = \\
 &= Pr(T(x_i, x_j) - g_{me,N}(x_i, x_j) \leq l) - \\
 &\quad Pr(T(x_i, x_j) - g_{me,N}(x_i, x_j) \leq l - 1) = \\
 &= \frac{N!}{(((N - 1)/2)!)^2} \int_{G(l-1)}^{G(l)} t^{(N-1)/2}(1 - t)^{(N-1)/2} dt, \tag{A4}
 \end{aligned}$$

where:

$$G(l) = P(T(x_i, x_j) - g_k(x_i, x_j) \leq l).$$

The expressions (A3) and (A4) are determined on the basis of beta distribution  $B(p, q)$ , with parameters  $p = q = (N + 1)/2$ . The expected value and variance of the distribution assume the form – respectively:  $\frac{1}{2}$  and  $((N + 1)/2)^2 / (N + 1)^2 (N + 2) = \frac{1}{4}(N + 2)$ . The variance of the distribution converges to zero for  $N \rightarrow \infty$  and the integrand in integrals (A3), (A4) is symmetric around  $\frac{1}{2}$ . These facts guarantee, that the distributions of the random variables:  $T(x_i, x_j) - g_{me,N}(x_i, x_j)$  ( $(x_i, x_j) \in \mathbf{X} \times \mathbf{X}$ ) are for each  $N$  unimodal, their probability functions assume maximum in zero (i.e. for  $T(x_i, x_j) - g_{me,N}(x_i, x_j) = 0$ ) and satisfy the inequalities (A1), (A2). Last two conditions are sufficient (see the assumptions (9), (12) and Lemma 1 in Section 3) for the inequality (38).

*Proof of the inequality (39).*

The inequality (39) is proved on the basis of Chebyshev inequality for expected value. The left-hand side of the inequality can be transformed to the form  $Pr(W_{me,N}^* < \tilde{W}_{me,N}) = 1 - Pr(W_{me,N}^* - \tilde{W}_{me,N} \geq 0)$  and each random variable  $Q_{ij,me,N}^{(\nu)}$  is transformed to the form, which provides non-negative expected value:

$$Q'_{ij,me,N}^{(\nu)} = Q_{ij,me,N}^{(\nu)} + (m - 1) \quad (\nu = 1, 2, 3, 6), \tag{A5}$$

$$Q'_{ij,me,N}^{(\nu)} = Q_{ij,me,N}^{(\nu)} + 2(m - 1) \quad (\nu = 5, 8), \tag{A6}$$

$$Q'_{ij,me,N}^{(\nu)} = Q_{ij,me,N}^{(\nu)} + (m - 2) \quad (\nu = 4, 7). \tag{A7}$$

After transformations (A5)-(A7), the probability  $Pr(W_{me,N}^* - \tilde{W}_{me,N} \geq 0)$  can be evaluated in the following way:



$$\begin{aligned}
 Pr(W_{me,N}^* - \tilde{W}_{me,N} \geq 0) &= P\left(\sum_{\nu=1}^8 \sum_{S_\nu} Q_{ij,me,N}^{(\nu)} \geq 0\right) = \\
 Pr\left(\sum_{\nu=1}^8 \sum_{S_\nu} Q_{ij,me,N}'^{(\nu)} \geq \lambda_1(m-1) + 2\lambda_2(m-1) + \lambda_3(m-2)\right) &\leq \\
 &\leq \frac{E\left(\sum_{\nu=1}^8 \sum_{S_\nu} Q_{ij,me,N}'^{(\nu)}\right)}{\lambda_1(m-1) + 2\lambda_2(m-1) + \lambda_3(m-2)} = \\
 &= 1 + \frac{E\left(\sum_{\nu=1}^8 \sum_{S_\nu} Q_{ij,me,N}^{(\nu)}\right)}{(\lambda_1 + 2\lambda_2)(m-1) + \lambda_3(m-2)}. \tag{A8}
 \end{aligned}$$

The inequality (A8) is equivalent to proved inequality (39).

---

# Feature Selection for High-Dimensional Data – A Pearson Redundancy Based Filter

Jacek Biesiada<sup>1</sup> and Włodzisław Duch<sup>2</sup>

<sup>1</sup> Division of Computer Methods, Dept. of Electrotechnology, The Silesian University of Technology, Katowice, Poland

<sup>2</sup> Dept. of Informatics, Nicolaus Copernicus University, Toruń, Poland  
Jacek.Biesiada@polsl.pl

**Summary.** An algorithm for filtering information based on the Pearson  $\chi^2$  test approach has been implemented and tested on feature selection. This test is frequently used in biomedical data analysis and should be used only for nominal (discretized) features. This algorithm has only one parameter, statistical confidence level that two distributions are identical. Empirical comparisons with four other state-of-the-art features selection algorithms (FCBF, CorrSF, ReliefF and ConnSF) are very encouraging.

## 1 Introduction

For large highly dimensional datasets feature ranking and feature selection algorithms are usually of the filter type [1]. In the simplest case feature filter is a function (such as correlation or information content) returning a relevance index  $J(\mathcal{S}|\mathcal{D}, C)$  that estimates, given the data  $\mathcal{D}$ , how relevant a given feature subset  $\mathcal{S}$  is for the task  $C$  (usually classification or approximation of data). An algorithmic procedure, such as building a decision tree or finding nearest neighbors, may also be used to estimate this index. The  $J(\mathcal{S}|\mathcal{D}, C)$  filter index is calculated directly from data, without any reference to the results of programs that are used for final data analysis. Since the data  $\mathcal{D}$  and the task  $C$  are usually fixed and only the subsets  $\mathcal{S}$  varies an abbreviated form  $J(\mathcal{S})$  will be used.

Relevance indices computed for individual features  $X_i, i = 1 \dots N$  establish a ranking order  $J(X_{i_1}) \leq J(X_{i_2}) \dots \leq J(X_{i_N})$ . Those features which have the lowest ranks may be filtered out. For independent features this may be sufficient, but if features are correlated many of them may be redundant. Moreover, for some data distributions the best pair of features may not even include a single best feature [2]! Thus ranking does not guarantee that the largest subset of important features will be found. Methods that search for the best subset of features may also use filters to evaluate the usefulness of subsets of features.

The thresholds for feature rejection may be set either for relevance indices, or by evaluation of reduced dimensionality results. Features are ranked by the filter, but how many are finally taken may be determined using adaptive system as a wrapper. Evaluation of the adaptive system performance (usually crossvalidation tests) are done only for a few pre-selected feature sets, but still this “frapper”

(filter-wrapper) approach may be rather costly if many feature subsets are evaluated. What is needed is a simple filter method that may be applied to a large datasets ranking and removing redundant features, parameterized in statistically well-established way. Such an approach is described in this paper. Similar filter for reducing redundant continuous features based on Kolmogoros-Smirnov test has been proposed in [3].

In the next section relevance index based on Pearson’s  $\chi^2$  test to estimate correlation between the distribution of feature values and the class labels is introduced. Section 3 compares it with four state-of-the-art feature selection algorithms using three bioinformatics datasets.

## 2 Relevance Indices and Algorithms

### 2.1 Correlation-Based Measures

For feature  $X$  with values  $x$  and classes  $C$  with values  $c$ , where  $X, C$  are treated as random variables, Pearson’s linear correlation coefficient is defined as [4]:

$$\varrho(X, C) = \frac{E(XC) - E(X)E(C)}{\sqrt{\sigma^2(X)\sigma^2(C)}} = \frac{\sum_i (x_i - \bar{x}_i)(c_i - \bar{c}_i)}{\sqrt{\sum_i (x_i - \bar{x}_i)^2 \sum_j (c_j - \bar{c}_j)^2}}. \quad (1)$$

$\varrho(X, C) = \pm 1$  if  $X$  and  $C$  are linearly dependent and zero if they are completely uncorrelated. Probability that two variables are correlated is estimated using the error function [4]  $\mathcal{P}(X \sim C) = \text{erf}\left(|\varrho(X, C)|\sqrt{N/2}\right)$ . The feature list ordered by decreasing values of the  $\mathcal{P}(X \sim C)$  may serve as feature ranking. An alternative approach is to use  $\chi^2$  statistics, but in both cases for large number of samples probability  $\mathcal{P}(X \sim C)$  is so close to 1 that ranking becomes impossible due to the finite numerical accuracy of computations. With  $N = 1000$  samples coefficients as small as  $\varrho(X, C) \approx 0.02$  give correlation probabilities  $\mathcal{P}(X \sim C) \approx 0.5$ . The  $\varrho(X, C)$  or  $\chi^2$  thresholds for the significance of a given feature may therefore be taken from a large interval corresponding to almost the same probabilities of correlation. Non-parametric, or Spearman’s rank correlation coefficients is useful for ordinal data types.

Information theory is frequently used to define relevance indices. The Shannon information for distribution of feature values and classes is:

$$H(X) = - \sum_i \mathcal{P}(x_i) \log \mathcal{P}(x_i); \quad H(C) = - \sum_i \mathcal{P}(c_i) \log \mathcal{P}(c_i) \quad (2)$$

and the joint Shannon entropy is:

$$H(X, C) = - \sum_{i,j} \mathcal{P}(x_i, c_j) \log \mathcal{P}(x_i, c_j) \quad (3)$$

Information filtering is frequently based on mutual information (MI):

$$MI(X, C) = H(X) + H(C) - H(X, C) \quad (4)$$

or on the Symmetrical Uncertainty Coefficient (SU) with similar properties:

$$SU(X, C) = 2 \frac{MI(X, C)}{H(X) + H(C)} \quad (5)$$

If a group of  $k$  features  $\mathbf{X}_k$  has already been selected, correlation coefficient may be used to estimate correlation between this group and the class, including inter-correlations between the features. Denoting the average correlation coefficient between these features and classes as  $r_{kc} = \bar{\varrho}(\mathbf{X}_k, C)$  and the average between different features as  $r_{kk} = \bar{\varrho}(\mathbf{X}_k, \mathbf{X}_k)$  the relevance of the feature subset is defined as:

$$J(\mathbf{X}_k, C) = \frac{kr_{kc}}{\sqrt{k + (k-1)r_{kk}}}. \quad (6)$$

This formula has been used in the Correlation-based Feature Selection (CFS) algorithm [5] adding (forward selection) or deleting (backward selection) one feature at a time. A definition of predominant correlation proposed by Yu and Liu [6] for Fast Correlation-Based Filter (FCBF) includes correlations between feature and classes and between pairs of features. The FCBF algorithm does a typical ranking using  $SU$  coefficient (Eq. 5) to determine class-feature relevance, setting some threshold value  $SU \geq \delta$  or number of features  $\lfloor n \log(n) \rfloor$  to determine how many features should be taken. In the second part redundant features are removed by defining the “predominant features”.

Selection method called ConnSF, based on inconsistency measure, has been proposed by Dash *et al.* [7] and will be used for comparison in Sec. 3. Two identical input vectors are inconsistent if they have identical class labels (a similar concept is used in rough set theory). Intuitively it is clear that inconsistency grows when the number of features is reduced and that feature subsets that lead to high inconsistency are not useful. If there are  $n$  samples in the dataset with identical feature values  $x_i$ , and  $n_k$  among them belong to class  $k$  then the inconsistency count is defined as  $n - \max_k c_k$ . The total inconsistency count for a feature subset is the sum of all inconsistency counts for all data vectors.

A different way to find feature subsets is used in the Relief algorithm [8]. This algorithm estimates weights of features according to how well their values distinguish between data vectors that are near to each other. For a randomly selected vector  $X$  from a data set  $\mathcal{S}$  with  $k$  features Relief searches the dataset for its two nearest neighbors: the nearest hit  $H$  from the same class and the nearest miss  $M$  from another class. For feature  $x$  and two input vectors  $X, X'$  the contribution to the weight  $W_x$  is proportional to the  $D(x, X, X') = 1 - \delta(X(x), X'(x))$  for binary or nominal features, and  $D(x, X, X') = |X(x) - X'(x)|$  for continuous features. The process is repeated  $m$  times, where  $m$  is a user defined parameter. Normalization with  $m$  in calculation of  $W_x$  guarantees that all weights are in the  $[-1, 1]$  interval. In Sec. 3 an extension of this algorithm for multiclass problems, called ReliefF [8] has been used.

## 2.2 Pearson’s Redundancy Based Filters

The Pearson  $\chi^2$  test measures the difference between the probability distribution of two binned random variables. If a feature is redundant than the hypothesis that its distribution is equal to already selected feature should have high probability.  $n$  independent observations of two random variables  $X, X'$  are given in the training data, where for the Pearson  $\chi^2$  test to be valid  $n$  should be more than 100. The test for  $X, X'$  feature redundancy proceeds as follows:

- Frequencies  $f_i, f'_i$  of occurrences of feature values in each bin are recorded (counting unique feature values).
- Based on the frequency counts empirical probability distributions  $F_i$  and  $F'_i$  are constructed and  $\chi^2(X, X')$  matrix is constructed:

$$\chi^2(X, X') = \sum_{i=1}^k \frac{(F_i - F'_i)^2}{F'_i} \tag{7}$$

A large value of  $\chi^2$  or a different number of unique feature values indicates that features are not redundant. When p-value  $p(\chi^2) > \alpha$  then the two distributions are equivalent with  $\alpha$  significance level, and thus one of the features is redundant. The best p-value could be estimated independently for each classifier using crossvalidation techniques. Below several estimates for different values of  $\alpha$  are made to find the optimal value for each classification method. This represents the frapper approach of using filter for ranking and adding wrapper in the final determination of the number of selected features.

Pearson’s Redundancy Based Filter (PRBF) algorithm is presented in Fig. 1 First, the relevance is determined using the symmetrical uncertainty (other relevance criteria may also be used), and then  $\chi^2$  test is applied to remove redundancy.

---

**Algorithm PRBF:**

Relevance analysis

1. Calculate  $SU(X, C)$  relevance indices and create an ordered list  $\mathcal{S}$  of features according to the decreasing value of their relevance.

**Redundancy analysis**

2. Take as  $X$  the first feature from the  $\mathcal{S}$  list
  3. Find and remove all features for which  $X$  is approximately equivalent according to the Pearson  $\chi^2$  test
  4. Set the next remaining feature in the list as  $X$  and repeat step 3 for all remaining features in the  $\mathcal{S}$  list.
- 

**Fig. 1.** A two-step Pearson’s Redundancy Based Filter (PRBF) algorithm

## 3 Empirical Studies

To evaluate the performance of the PCBF algorithm both artificial and real datasets have been used with a number of classification methods. Two artificial

**Table 1.** Accuracy of 4 classifiers on selected subsets of features for the Gauss8 dataset

Title	Selected features					
	Full set	FCBF	CorrSF	ReliefF	ConnSF	PRBF
Features	1 to 8	1+2+3	1+2+5	1+5	1 to 4	1 to 4
C4.5	78.85 ± 0.36	79.21 ± 0.29	78.64 ± 0.31	76.15 ± 0.09	78.85 ± 0.36	78.85 ± 0.36
NBC	82.07 ± 0.07	81.57 ± 0.08	80.25 ± 0.07	76.98 ± 0.06	82.08 ± 0.07	82.07 ± 0.07
1NN	73.48 ± 0.25	73.57 ± 0.22	71.33 ± 0.25	68.19 ± 0.34	73.48 ± 0.25	73.48 ± 0.25
SVM	81.97 ± 0.08	81.54 ± 0.10	80.77 ± 0.07	76.98 ± 0.07	81.88 ± 0.08	81.87 ± 0.09
Average	79.09 ± 0.19	78.97 ± 0.17	77.75 ± 0.18	74.57 ± 0.14	79.07 ± 0.19	79.07 ± 0.20

**Table 2.** Summary of the datasets used in empirical studies

Title	Features	Instances	Classes
Lung-cancer (Lung)	58	32	3
Promoters	59	106	2
Splice	62	3190	3

datasets, Gauss4, and Gauss8, have been used in our previous study [9]. Gauss4 is based on sampling from 4 Gaussian functions with unit dispersion in 4 dimensions, each cluster representing a separate class. The first function is centered at  $(0, 0, 0, 0)$ , the next at  $(1, 1/2, 1/3, 1/4)$ ,  $(2, 1, 2/3, 1/2)$ , and  $(3, 3/2, 3, 3/4)$ , respectively. The dataset contains 4000 vectors, 1000 per each class. In this case the ideal ranking should give the following order:  $X_1 > X_2 > X_3 > X_4$ .

Gauss8 used here is an extension of Gauss4, adding 4 additional features that are approximately linearly dependent  $X_{i+4} = 2X_i + \epsilon$ , where  $\epsilon$  is a uniform noise with a unit variance. In this case the ideal ranking should give the following order:  $X_1 > X_5 > X_2 > X_6 > X_3 > X_7 > X_4 > X_8$  and the selection should reject all 4 linearly dependent features as redundant. The PRBF and the ConnSF [7] algorithms had no problem with this task, but FCBF [6] selected only 3 features, CorrSF [5] selected only first two, and ReliefF [8] left only feature 1 and 5, giving them both the same weight 0.154 (for features 2 and 6 the weight was 0.060, dropping to 0.024 for feature 3, 6 and to 0.017 for features 4, 8).

In Table 1 results of Naive Bayes Classifier (NBC) (Weka implementation, [10]), the nearest neighbor algorithm (1NN) with Euclidean distance function, C4.5 tree [12] and the Support Vector Machine with a linear kernel are given (Weka and SVM, Ghostminer 3.0 implementation<sup>1</sup>).

For the initial comparison on real data three biomedical datasets from the UCI Machine Learning Repository [11] were used. A summary of all datasets is presented in Table 2. They have rather modest number of nominal features and range from 32 to 3190 samples. Lungs dataset is extremely small and 5 out of 32 instances containing missing values have been removed. The purpose is to see the influence of the number of samples on the quality of results for similar number of nominal features.

For each data set all five feature selection algorithms are compared (FCBF [6], CorrSF [5], ReliefF [8], ConnSF [7], and PRBF) and the number of features

<sup>1</sup> <http://www.fqspl.com.pl/ghostminer/>

selected by each algorithm is given. 5 neighbors, 30 instances and treshold 0.1 were used for ReliefF, as suggested by Robnik-Sikonja and Kononenko [8]. For CorrSF and ConnSF forward search strategy has been used, and for FCBF, ReliefF, and the PRBF forward search strategy based on ranking.

**Table 3.** The number of selected features for each algorithm; bold face – lowest number, italics – highest number

Dataset	Selected features					
	Full set	FCBF	CorrSF	ReliefF	ConnSF	PRBF
Lung-cancer	58	6	7	11	<b>4</b>	<i>12</i>
Splice	62	22	<b>6</b>	<i>24</i>	10	19
Promoters	59	6	<b>4</b>	<i>12</i>	<b>4</b>	6
Average	59.6	11.3	<b>5.6</b>	<i>15.6</i>	6	12.2

In Table 4 results of Naive Bayes Classifier (NBC) (Weka implementation, [10]), the nearest neighbor algorithm (1NN) with Euclidean distance function, C4.5 tree [12] and the Support Vector Machine with a linear kernel and  $C = 1$  (estimated to be close to optimal value for these datasets) are collected. The overall average balanced accuracy (accuracy for each class, averaged over all classes) and the standard deviation obtained from averaging 20 repetitions of 10-fold cross-validation calculations with different initializations is reported in Tables below. For datasets with significant differences in *a priori* class distributions balanced accuracy is more sensitive measure than the overall accuracy.

**Table 4.** Balanced accuracy for the 4 classification methods on features selected by each algorithm; bold face – best results, italics – worst

C 4.5 tree						
Method	Full set	FCBF	CorrSF	ReliefF	ConnSF	<i>PRBF<sub>0.001</sub></i>
Lung	<b>80.52 ± 3.53</b>	<i>76.30 ± 2.88</i>	<b>80.52 ± 3.53</b>	<b>80.52 ± 3.53</b>	<b>80.52 ± 3.53</b>	<i>77.37 ± 3.49</i>
Splice	<i>94.16 ± 0.26</i>	<b>94.30 ± 0.24</b>	<i>93.07 ± 0.16</i>	<i>94.02 ± 0.19</i>	<i>93.83 ± 0.21</i>	<i>94.03 ± 0.22</i>
Promoters	<i>79.20 ± 1.90</i>	<i>81.04 ± 1.81</i>	<i>80.85 ± 2.65</i>	<i>81.09 ± 2.06</i>	<i>80.47 ± 2.21</i>	<b>82.69 ± 1.57</b>
Naive Bayes						
Method	Full set	FCBF	CorrSF	ReliefF	ConnSF	<i>PRBF<sub>0.001</sub></i>
Lung	<i>61.27 ± 4.67</i>	<i>87.37 ± 2.10</i>	<b>90.98 ± 1.95</b>	<i>83.43 ± 2.55</i>	<i>71.28 ± 3.93</i>	<i>88.09 ± 1.96</i>
Splice	<i>94.95 ± 0.08</i>	<b>96.10 ± 0.06</b>	<i>93.32 ± 0.05</i>	<i>95.54 ± 0.08</i>	<i>94.30 ± 0.08</i>	<i>94.62 ± 0.08</i>
Promoters	<i>90.47 ± 1.40</i>	<i>94.43 ± 0.52</i>	<b>94.58 ± 0.86</b>	<i>91.27 ± 1.18</i>	<i>92.45 ± 1.30</i>	<i>91.18 ± 0.93</i>
1 Nearest Neighbor						
Method	Full set	FCBF	CorrSF	ReliefF	ConnSF	<i>PRBF<sub>0.001</sub></i>
Lung	<i>47.55 ± 5.61</i>	<i>78.83 ± 2.98</i>	<b>82.17 ± 4.23</b>	<i>78.59 ± 3.71</i>	<i>74.33 ± 5.11</i>	<i>70.60 ± 5.02</i>
Splice	<i>80.16 ± 0.47</i>	<i>85.14 ± 0.44</i>	<i>84.60 ± 2.19</i>	<i>83.54 ± 0.44</i>	<b>87.13 ± 0.64</b>	<i>84.37 ± 0.65</i>
Promoters	<i>81.27 ± 2.40</i>	<i>85.24 ± 2.51</i>	<b>88.63 ± 1.90</b>	<i>81.04 ± 1.81</i>	<i>85.38 ± 2.62</i>	<i>85.33 ± 3.02</i>
SVM						
Method	Full set	FCBF	CorrSF	ReliefF	ConnSF	<i>PRBF<sub>0.001</sub></i>
Lung	<i>47.90 ± 3.71</i>	<i>84.48 ± 2.74</i>	<b>90.00 ± 0.00</b>	<b>90.00 ± 0.00</b>	<i>80.63 ± 2.07</i>	<i>80.78 ± 2.07</i>
Splice	<i>92.35 ± 0.31</i>	<b>95.78 ± 0.15</b>	<i>93.74 ± 0.03</i>	<i>95.49 ± 0.24</i>	<i>94.24 ± 0.16</i>	<i>94.99 ± 0.17</i>
Promoters	<i>91.51 ± 1.65</i>	<i>93.68 ± 1.15</i>	<b>95.76 ± 0.94</b>	<i>87.78 ± 2.38</i>	<i>87.31 ± 1.08</i>	<i>90.66 ± 1.96</i>

In Table 5 classification results for various significance levels are presented. Surprisingly the best results have been obtained for a very small level  $\alpha = 0.001$ , removing the largest number of redundant features.

**Table 5.** Number of features for different levels of significance, and balanced accuracy (bacc)  $\pm$  std(bacc) for C4.5, NBC, 1NN and SVM classifiers

$\alpha$	0.001	0.01	0.05	0.1	0.15	0.2
Lung	12	14	16	16	18	20
C4.5	77.37 $\pm$ 3.49	77.37 $\pm$ 3.49	77.37 $\pm$ 3.49	77.37 $\pm$ 3.49	77.37 $\pm$ 3.49	77.37 $\pm$ 3.49
NBC	88.09 $\pm$ 1.96	85.56 $\pm$ 1.14	83.22 $\pm$ 2.45	83.22 $\pm$ 2.45	84.89 $\pm$ 1.85	84.47 $\pm$ 2.04
1NN	70.60 $\pm$ 5.02	72.17 $\pm$ 4.64	68.49 $\pm$ 3.72	68.49 $\pm$ 3.75	65.88 $\pm$ 3.61	63.69 $\pm$ 4.37
SVM	80.78 $\pm$ 2.07	76.45 $\pm$ 3.33	75.08 $\pm$ 3.27	75.08 $\pm$ 3.27	72.16 $\pm$ 3.09	70.20 $\pm$ 4.04
Splice	19	24	27	28	30	31
C4.5	94.03 $\pm$ 0.22	94.03 $\pm$ 0.22	94.19 $\pm$ 0.21	94.19 $\pm$ 0.21	94.19 $\pm$ 0.21	94.22 $\pm$ 0.20
NBC	94.62 $\pm$ 0.08	94.62 $\pm$ 0.08	95.11 $\pm$ 0.11	95.08 $\pm$ 0.07	94.96 $\pm$ 0.10	95.25 $\pm$ 0.07
1NN	84.37 $\pm$ 0.65	84.37 $\pm$ 0.65	81.40 $\pm$ 0.48	80.46 $\pm$ 0.58	80.66 $\pm$ 0.42	81.14 $\pm$ 0.41
SVM	94.99 $\pm$ 0.17	95.00 $\pm$ 0.17	94.49 $\pm$ 0.22	94.44 $\pm$ 0.19	94.20 $\pm$ 0.17	94.42 $\pm$ 0.22
Promoters	6	8	11	13	13	14
C4.5	82.69 $\pm$ 1.57	82.41 $\pm$ 1.69	79.72 $\pm$ 2.09	79.77 $\pm$ 1.72	79.77 $\pm$ 1.72	79.53 $\pm$ 1.73
NBC	91.18 $\pm$ 0.93	91.98 $\pm$ 0.94	92.78 $\pm$ 1.24	91.65 $\pm$ 0.98	91.65 $\pm$ 0.98	92.45 $\pm$ 0.69
1NN	85.33 $\pm$ 3.02	85.10 $\pm$ 2.90	88.68 $\pm$ 1.81	86.13 $\pm$ 2.37	86.13 $\pm$ 2.37	85.33 $\pm$ 2.18
SVM	90.66 $\pm$ 1.96	90.09 $\pm$ 2.09	86.93 $\pm$ 2.04	87.88 $\pm$ 1.45	87.88 $\pm$ 1.45	88.35 $\pm$ 2.04

## 4 Conclusion

A new algorithm for finding non-redundant binned feature subsets based on the Pearson  $\chi^2$  test has been introduced. PRBF has only one parameter, statistical significance or the probability that the hypothesis that distributions of two features is equivalent is true. In the first step SU indices Eq. 5 have been used for ranking, and in the second step redundant features are removed in an unsupervised way, because during reduction of redundant features information about the classes is not used. Our initial tests are encouraging: on the artificial data perfect ranking has been recreated and redundant features rejected, while on the real data, with rather modest number of features selected results are frequently the best, or close to the best, comparing with four state-of-the-art feature selection algorithms. The new algorithm seems to work especially well with the linear SVM classifier. Computational demands of PRBF algorithm are similar to other correlation-based filters, and much lower than ReliefF. The  $\chi^2$  test works well for  $n > 100$  samples, therefore the results for very small Lung-cancer data are rather poor.

For simplicity of interpretation only data with nominal features have been used, avoiding discretization. Features were ranked according to the SU relevance index. In real applications with very large number of features a cutoff point for ranking should be defined and optimized using crossvalidation tests to determine optimal threshold value. Further reduction of the selected feature subsets using tests for redundancy requires another estimation of the significance parameter that may be done in crossvalidation test and will depend on classifier used. Such frapper (filter-wrapper) approach is not too costly and may be completely automatic. The same algorithm may be used with other indices for relevance indication. Moreover, redundancy reduction based on  $\chi^2$  test may be used in unsupervised methods of data analysis. Various variants of this and similar test exist [4], including versions for small samples. This combination of



filters, wrappers and redundancy evaluation is a fertile ground for information selection, with many possibilities that remain to be explored. Further tests on much larger bioinformatics data will be reported soon.

*Acknowledgement.* This work was financed by the Polish Committee for Scientific Research grant 2005-2007 to WD; JB has been supported by the Polish Foundation of Science and grant (2006-2009) No.: 6ZR9 2006 C/06742.

## References

1. W. Duch, *Filter Methods*. In: Feature extraction, foundations and applications. Eds: I. Guyon, S. Gunn, M. Nikravesh, L. Zadeh, Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer, pp. 89-118, 2006.
2. T.M. Cover. The best two independent measurements are not the two best. *IEEE Transactions on Systems, Man, and Cybernetics*, 4:116-117, 1974.
3. J. Biesiada, W. Duch, Feature Selection for High-Dimensional Data: A Kolmogorov-Smirnov Correlation-Based Filter Solution. *Advances in Soft Computing, Computer Recognition Systems (CORES 2005)*, pp. 95-105, 2005.
4. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C. The art of scientific computing*. Cambridge University Press, Cambridge, UK, 1988.
5. M.A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, Department of Computer Science, University of Waikato, Waikato, N.Z, 1999.
6. L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *12th Int. Conf. on Machine Learning (ICML-03), Washington, D.C.*, pp. 856-863, Morgan Kaufmann, CA 2003.
7. M. Dash and H. Liu. Consistency-based search in feature selection. *Artificial Intelligence*, 151:155-176, 2003.
8. M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of relief and rrelieff. *Machine Learning*, 53:23-69, 2003.
9. W. Duch, T. Winiarski, J. Biesiada, and A. Kachel. Feature ranking, selection and discretization. In *Proceedings of Int. Conf. on Artificial Neural Networks (ICANN)*, pages 251-254, Istanbul, 2003. Bogazici University Press.
10. I. Witten and E. Frank. *Data mining - practical machine learning tools and techniques with JAVA implementations*. Morgan Kaufmann, San Francisco, CA, 2000.
11. C.J. Mertz and P.M. Murphy. *The UCI repository of machine learning databases*. Univ. of California, Irvine, 1998.  
<http://www.ics.uci.edu/pl/mlearn/MLRespository.html>.
12. J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, CA, 1993.

---

# Some Approximation Algorithms for Minimum Vertex Cover in a Hypergraph

Barbara Maźbic-Kulma and Krzysztof Sep

System Research Institute PAS, ul. Newelska 6, 01-447 Warsaw  
{kulma,sep}@ibspan.waw.pl

**Summary.** The paper presents five algorithms for minimum vertex cover in a hypergraph. We consider: exact-backtracking, *LJC*, *SBT*, *RSBT* and *MSBT* algorithms. For a set of randomly generated hypergraphs tests have been performed to compare the aforementioned procedures.

## 1 Introduction

The ever growing need to store and process data of increasing volume requires suitable handling and analysis. Many types of data can be represented in the form of a hypergraph. Some problems in data management can be characterized as a minimum vertex cover in a hypergraph. The minimum cover procedure can have a role as a preprocessing routine. *SBT* algorithm for a minimum vertex cover in a hypergraph has been described in [9]. In what follows we develop its modifications (algorithms *RSBT* and *MSBT*). They belong to the approximation-type algorithms and as such they find only an approximation to the optimal solution. We demonstrate that in combination with the algorithm *LJC* [2, 6, 10] we are able to estimate the quality of the obtained solution. Tests show that by applying one of our algorithms one can obtain better results – depending on the type of hypergraph.

## 2 Basic Concepts

A **Hypergraph** is a pair  $H = (X, F)$ , where  $X$  is a non-empty, finite set of vertices,  $F$  is a non-empty family of different subsets of set  $X$  fulfilling the condition:  $f \in F \Rightarrow f = X$  where  $F$  is a set of edges.[1].

Note that an edge can contain any number of vertices (even one), which makes a difference between hypergraphs and graphs. In fact, hypergraphs can be viewed as a direct generalization of graphs.

**Degree** of a vertex is the number of edges to which this vertex belongs.

A **transversal**  $Tr$  of a hypergraph  $H = (X, F)$  is a subset of vertices which has the property  $\forall e \in E : e \cap Tr = \emptyset$ . In other words, a transversal is a set of vertices that covers (or blocks, hence the other name: *blocking set*) all the edges. A transversal  $T_{mi}$  is called minimum in the sense of inclusion if there exists no subset of  $T_{mi}$  being a transversal. A transversal is called minimum in the sense of cardinality if there exists no other transversal having less elements.

Minimum transversal problem is  $NP$ [2, 3, 5]. All known exact algorithms solving this problem are of exponential complexity so they are of no use.[7, 8], this begin the reason to use approximating algorithms. [2, 4, 6, 11].

### 3 Algorithms

As mentioned in the introduction we present five algorithms for a minimum vertex cover in a hypergraph i.e. a minimum transversal in a hypergraph. Below, the algorithms are described in turn: exact-backtracking, *Lovász-Johnson-Chvatal*, *SBT*, *RSBT*, *MSBT*.

#### 3.1 Exact Algorithms - Backtracking

This algorithm finds minimum cover but its worst-case number of steps to find a minimum transversal is  $O(2^{|X|})$ , and the expected number of steps is:  $E(X) = \sum_{k=0}^{|X|} \binom{|X|}{k} 2^{-2^k}$ . Thus it is practically useless in solving the real problems.

---

```

Require:  $C$  - the set of vertices and  $v$  - the first vertex to be removed from  $C$ ,
in the beginning,  $C = X$  and  $v = 0$ ;
for all  $w \in C$  such that  $w \geq v$  do
if  $C - \{w\}$  is a transversal and  $C - \{w\}$  wasn't already visited then
    Set  $C_{\min} := C$  if  $C$  has less elements than  $C_{\min}$  (the current best solution);
    Call the procedure recursively with parameters  $C := C - \{w\}$ ,  $v := w$ ;
end if
end for
    
```

---

#### 3.2 Approximating Algorithm of *Lovász-Johnson-Chvatal* (LJC)

The *Lovász-Johnson-Chvatal* algorithm is a greedy algorithm – in each step it chooses the vertex that covers the biggest number of edges. Using the *Lovász-Johnson-Chvatal* algorithm we assume that in set of vertices should contain these vertices which cover the biggest number of edges.[2, 6, 10, 12]

Let  $deg(x)$  be a degree of vertex and  $F(x)$  be a set of edges to which vertex  $x$  belongs. The *Lovász-Johnson-Chvatal* algorithm for hypergraph  $H = (X, F)$  works as follows:

---

```

procedure LJC(H)
begin
     $Tr := \emptyset$ ;  $V := X$ ;  $E := F$ ;
    while ( $E \neq \emptyset$ ) do
        begin
            choose vertex  $V : \forall z \in V \ deg(z) \leq deg(v)$ ;
             $V := V \setminus \{v\}$ ;
             $E := E \setminus F(v)$ ;
             $Tr := Tr \cup v$ ;
        end;
    end;
end;
    
```

---

Using the greedy algorithm *LJC* we obtain the transversal, which can have proper subset being a transversal. If  $n$  is cardinality of the vertex set and  $t_{opt}$  is cardinality of the smallest transversal and  $t_a$  is cardinality of the vertex set obtained using *LJC* algorithm then the following condition is satisfied [2, 6, 11]:  $t_{opt} \leq t_a \leq t_{opt}(1 + \lg_2 n)$ .

The *Lovász–Johnson–Chvatal* algorithm finds the transversal of a hypergraph, for some hypergraphs it can be a minimum transversal in sense of cardinality but there exist hypergraphs that *Lovász–Johnson–Chvatal* algorithm finds transversal which is not minimum in the sense of inclusion.

### 3.3 Approximation Algorithm *SBT*

The backtracking algorithm which is an exact algorithm, has exponential complexity, the *Lovász–Johnson–Chvatal* approximation algorithm has polynomial complexity but transversal obtained using this algorithm may be not transversal in the sense of inclusion.

The *SBT* algorithm searches for the vertices of the smallest degree and removes them from the set of vertices. If without the removed vertex the reduced vertex set (reduct) is not a transversal, then we add the removed vertex to the transversal under construction and the edges incident to this vertex are removed from the hypergraph – these edges are treated as covered. This routine will be repeated until all edges are removed i.e. until all edges are covered.

Let  $H = (X, F)$  be a hypergraph for which a minimum transversal is sought,  $X$  – the set of its vertices,  $F$  – the set of its edges.

Let  $m(x)$  be a vertex in  $X$  incident upon the minimum number of edges; if there are several such vertices we pick any of them. Let  $Z(X, F)$  be a set of vertices in the reduct of the hypergraph  $H(X, F)$ ; the elements of  $Z$  do not belong to any edge.

Let  $F(x)$  be a set of edges incident with the vertex  $x$ . We carry out the *SBT* algorithm as follows:

---

```

procedure SBT(H)
begin
   $Tr := \emptyset; V := X; Q := X; Q := X; E := F;$ 
  while  $V \neq \emptyset$  and  $E \neq \emptyset$  do
    begin
       $k := m(V);$ 
       $V := V \setminus \{k\};$ 
      if  $V$  is not transversal of hypergraph  $(Q, E)$  do
        begin
           $Tr := Tr \cup \{k\};$ 
           $E := E \setminus F(x);$ 
           $V := V \setminus Z(V, E);$ 
        end;
      end;
       $Q := V;$ 
    end;
  end;

```

---

### 3.4 Approximation Algorithm *RSBT*

The *RSBT* algorithm seeks the vertices with the highest degree and removes them from the set of vertices. If without a removed vertex the reduct is not a transversal then the removed vertex should be added to the transversal under construction and the edges incident with this vertex are removed from the hypergraph – they are deemed to be covered. This procedure will be repeated until all edges are removed i.e. until all edges are covered.

Let  $H = (X, F)$  be a hypergraph for which a minimum transversal is sought;  $X$  - the set of its vertices,  $F$  - the set of its edges.

Let  $m(x)$  be a vertex in  $X$  incident upon the minimum number of edges; if there are several such vertices we pick any of them. Let  $Z(X, F)$  be a set of vertices in the reduct of the hypergraph  $H(X, F)$ ; the elements of  $Z(X, F)$  do not belong to any edge.

Let  $F(x)$  be a set of all edges incident with the vertex  $x$ .

We carry out the *RSBT* algorithm as follows:

---

```

procedure RSBT(H)
begin
   $Tr := \emptyset$ ;  $V := X$ ;  $Q := X$ ;  $E := F$ ;
  while  $V \neq \emptyset$  and  $E \neq \emptyset$  do
    begin
       $k := M(V)$ ;
       $V := V \setminus \{k\}$ ;
      if  $V$  is not transversal of hypergraph  $(Q, E)$  do
        begin
           $Tr := Tr \cup \{k\}$ ;
           $E := E \setminus F(x)$ ;
           $V := V \setminus Z(V, E)$ ;
        end;
       $Q := V$ ;
    end;
  end;

```

---

### 3.5 Approximation Algorithm *MSBT*

The *MSBT* algorithm seeks the vertices with the lowest degree and removes them from the set of vertices. If without a removed vertex the reduct is not a transversal then the removed vertex should be added to the transversal under construction, and edges incident with this vertex are eliminated from the hypergraph - they are deemed to be covered. If without the removed vertex the reduct is a transversal then we have to search in it for all the vertices with the following property: in the reduct there is at least one edge covered by exactly that precise vertex. We remove these vertices, and all the edges covered by them from the reduct to the transversal we add these vertices. This procedure will be repeated until all edges are removed i.e. until all edges are covered.

Let  $H = (X, F)$  be a hypergraph for which a minimum transversal is sought;  $X$  - the set of its vertices,  $F$  - the set of its edges.

Let  $m(x)$  be a vertex in  $X$  incident upon the minimum number of edges; if there are several such vertices we pick any of them.

Let  $Z(X, F)$  be a set of vertices in the reduct of the hypergraph  $H(X, F)$ ; the elements of  $Z(X, F)$  do not belong to any edge. Let  $F(x)$  be a set of all edges incident with the vertex  $x$ . We carry out the *MSBT* algorithm as follows:

---

```

procedure MSBT(H)
begin
   $Tr := \emptyset; V := X; Q := X; E := F;$ 
  while  $V \neq \emptyset$  and  $E \neq \emptyset$  do
    begin
       $k := m(V);$ 
       $V := V \setminus \{k\};$ 
      if  $V$  is not transversal of hypergraph  $(Q, E)$  do
        begin
           $Tr := Tr \cup \{k\};$ 
           $E := E \setminus F(x);$ 
           $V := V \setminus Z(V, E);$ 
        end;
      else
        begin
          for each edge covers by exact one vertex  $v$ 
             $Tr := Tr \cup \{v\};$ 
             $E := E \setminus F(v);$ 
             $V := V \setminus \{v\};$ 
          end;
         $Q := V;$ 
      end;
    end;
  end;

```

---

The transversal obtained in the *SBT*, *RSBT* and *MSBT* algorithms is minimum in the sense of inclusion.

### 3.6 Comparison of the Algorithms

In section 3.2 we have noticed that the *LJC* algorithm can not always find a minimum transversal in the sense of inclusion.

There exist hypergraphs for which the *LJC* algorithm finds a transversal which is redundant.

We show an example of this kind of hypergraph in the Fig. 1.

Using the *LJC* algorithm we obtain: 1 step vertex no 1 ( $\deg(v_1) = 4$ ) then our  $tr = \{1\}$  and the reduced hypergraph we show in the Fig. 2.

Because all vertices are necessary we obtain  $tr = \{1, 2, 3, 4, 5\}$ .

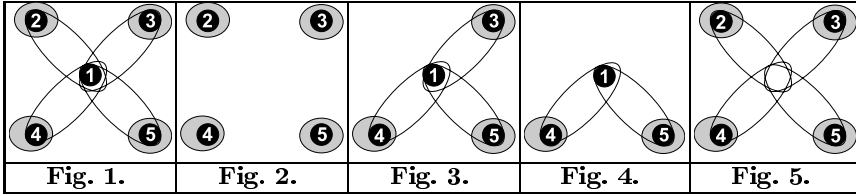
Using the *SBT* algorithm we remove vertex with the minimum degree, for example 2, but vertex 2 is necessary so our  $tr = \{2\}$  and the reduced hypergraph we show in the Fig. 3.

In the next step we remove 3, a necessary vertex as well, so our  $tr = \{2, 3\}$  and the reduced hypergraph we show in the Fig. 4.

All vertices have  $\deg = 2$ . If we remove vertex 1 our  $tr = \{2, 3\}$  because 1 is unnecessary and vertices 4 and 5 are necessary so our  $tr = \{2, 3, 4, 5\}$ .

If we use the MSBT algorithm we remove all vertices for which there exist edges covered only by these particular vertices so that our  $tr = \{2, 3, 4, 5\}$ .

If we use the RSBT algorithm we remove the vertex with the maximum degree and if it is necessary vertex we add it to our transversal. Vertex 1 is unnecessary so our hypergraph we show in the Fig. 5. and our  $tr = \{2, 3, 4, 5\}$ .



In the above mentioned particular case the algorithms *SBT*, *RSBT* and *MSBT* find a minimum transversal not only in the sense of inclusion but also in the sense of cardinality. The hypergraph described in the example above supports the argument that it is worth using all the approximation algorithms presented in the paper. Thanks to their polynomial complexity  $O(n^2)$ , for large hypergraphs the combined method is more effective than the exact algorithm.

### 4 Test Results

Some 20 000 tests have been run for randomly generated hypergraphs. Entries in the table bellow show the number of cases when individual algorithms could attain the minimum of all the transversals obtained.

SBT	LJC	MSBT	RSBT
4089	16083	4089	665

In practical problems an important question is to get not only a minimum cover but also a cover fulfilling some conditions on the sum of vertex degrees in the transversal. We have introduced an additional criterion – the sum of vertex degrees in the transversal – and in our tests we are counting the instances with transversals, obtained in our procedures, best fitted to our criterion. The results are presented in the table bellow.

	SBT	LJC	MSBT	RSBT
<i>fc min</i>	293	15318	293	666
<i>fc max</i>	5991	15	5991	10106

in which:

*fc min* corresponds to the condition of minimizing the sum of the vertex degrees;  
*fc max* corresponds to the condition of minimizing the sum of the vertex degrees;

To draw a conclusion from the results demonstrated in tables above we may state that one can not decide unambiguously which approximation algorithm gives the best solution. Therefore we suggest the following approach.

Let:

$H = (X, F)$  - be a hypergraph

---



---

$Tr_1$  := transversal found by algorithm *LJC*  
 $Tr_2$  := transversal found by algorithm *SBT*  
 $Tr_3$  := transversal found by algorithm *RSBT*  
 $Tr_4$  := transversal found by algorithm *MSBT*

---

From the transversals  $Tr_1, Tr_2, Tr_3$  and  $Tr_4$  choose the one which is minimum with respect to the criterion *fc min* or *tr min* or a maximum transversal with respect to the *fc max* criterion. If a chosen transversal has been obtained by the *LJC* algorithm, we may try to minimize it while applying one of the *SBT, RSBT* or *MSBT* algorithms. As a result we get a transversal minimal in the sense of inclusion.

With the above approach the estimation on the quality of the approximation is obtained as:

$$t_{opt} \leq t_a \leq t_{opt}(1 + \lg_2 n).$$

## 5 Conclusions

In this paper we show, that it cannot be unambiguously determined, which of the presented approximating algorithms finds a better solution. In special cases the *SBT, RSBT* and *MSBT* algorithms determine the minimal transversal regarding inclusion but also cardinality. The executed tests performed show the usefulness of using the all of above presented approximating algorithms. The polynomial complexity  $O(n^2)$  of the listed approximating algorithms results in the increased cost efficiency of the complex method for larger hypergraphs vs. using of the exact algorithm.

## References

1. Berge C, (1989) Hypergraphs, Combinatorics of Finite Sets North-Holland
2. Chvatal V, (1979) A greedy heuristic for the set-covering problem Mathematics and Operations Research 4
3. Cowen L, (1998) Approximation algorithms, Hopkins University
4. Feige U, (1998) A Threshold of  $\ln(n)$  for Approximating Set Cover Journal of the ACM
5. Hochbaum D S (ed.), (1997) Approximation Algorithms for NP-hard Problems PWS Publishing Company
6. Johnson D S (1974) Approximation Algorithms for Combinatorial Problems Journal of Computer and System Sciences 9



7. Jukna S, (2001) Extremal Combinatorics with application computer science Springer
8. Korte B, Vygen J, (2000) Combinatorial optimization, theory and algorithms, Springer
9. Kułaga P, Sapiecha P, Sęp K, (2005) Approximation Algorithm for the Argument Reduction Problem, Advances in soft computing Computer recognition systems CORES'05, Springer Verlag, Berlin
10. Lovasz L, (1975) On the ratio of optimal integral and fractional covers, Discrete Mathematics, 13
11. Maźbic-Kulma B, Sęp K, (2006) Baza wierzchołkowa w hipergrafie jako metoda przydzielania zadań, In: BOS
12. Sapiecha P, Selvaraj H, Stańczak J, Sęp K, Luba T, (2004) A hybrid approach to a classification, Intelligent information processing and web mining, Springer Verlag, Berlin

---

# Reference Set Reduction Algorithms Based on Double Sorting

Marcin Raniszewski

Technical University of Lodz, Computer Engineering Department, 90-924 Lodz, ul. Stefanowskiego 18/22  
mranisz@kis.p.lodz.pl

**Summary.** The reference set reduction algorithm for 1-NN rule, resulting in the consistent reduced reference sets, is presented. The algorithm uses well known Hart's algorithm. Samples from the reference set are initially sorted by two keys. The first one is the largest number of nearest neighbours from the same class, while the second one is a mutual distance measure introduced by Gowda and Krishna. The simple artificial example is used to show how the proposed algorithm operates. Two sets known as phoneme and satimage data have been used to verify the algorithm effectiveness, which is compared with the one offered by Gowda-Krishna algorithm.

## 1 Introduction

Reference set reduction algorithms for the 1-NN rule [3], [4] are constructed to eliminate the necessity to keep a large number of feature vectors in the computer memory. Usually, the reduced reference set ought to be consistent with the original one, i.e. the 1-NN operating with the reduced set classifies correctly all samples not included in the reduced set. However, some other criteria can be applied instead of the consistency.

The presented algorithm produces the consistent reduced reference set. The author's aim was even more far-reaching reduction than that promised by Gowda-Krishna algorithm [2], [3], [4] without increase of classification error rate. Both, mentioned already algorithms are described in details in the next section.

## 2 Hart's and Gowda-Krishna Algorithms

Hart's algorithm starts with the empty reduced reference set. The randomly chosen sample from reference set is added to the current reduced reference set. All the remaining samples from reference set (in random sequence) are classified by 1-NN rule operating with the current reduced reference set, which increases by each misclassified point. In one cycle of the algorithm all points from reference set, which are not yet in reduced reference set, are analyzed. These cycles are repeated until no points will be added to reduced reference set - then the algorithm is finished [1], [3], [4].

As we can see, in the first phase of Hart's procedure, random points are selected to reduced reference set. Hence, the result is dependent on primary arrangement of points in the original set. Thus, the samples selected as the first are not the most representative ones.

Gowda-Krishna algorithm removes this disadvantage only partially. The points from reference set are sorted by growing values of the mutual distance measure ( $mdm$ ) and then the Hart's procedure is applied. The mutual distance measure is calculated in the following way: for the point  $x$ , the nearest point  $y$  from opposite class is found. The mutual distance measure is the number of points from the same class as point  $x$ , which are closer to point  $y$  than to point  $x$ .

This kind of sorting seems to be accurate, because the points with a small mutual distance measure lie near the class borders and they should be put to the reduced set as the first.

### 3 The Analyze of the Consistent Reduced Reference Set Minimization Problem

Analysing the results obtained for large reference sets, one can conclude that numerous points exist with the same mutual distance measure. Therefore, despite of sorting, there will be still many points in the group with the same mutual distance measure without any established order. These points should be somehow sorted, preferably in the way, which results in a smaller consistent reduced reference set.

#### 3.1 Double Sorting

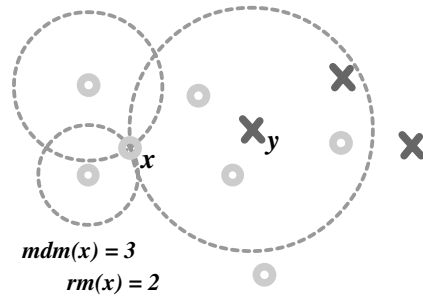
The samples that lie near the class borders should be included to reduced reference set. Additionally, we should add points, which represents well its class. The point represents its class well when for large number of the same class points it is closer (let's denote this number by  $rm$ ) than to any other point from the opposite class. The value  $rm$  will be called the representative measure of the point. The points should be sorted by decreasing values of the representative measures.

The points from the reference set can be sorted with the use of two keys,  $rm$  and  $mdm$  (fig. 1), in two manners:

1. The points are sorted first by mutual distance measure and then the points with the same  $mdm$  are arranged by the representative measure;
2. The points are sorted by decreasing representative measure and in the groups of points with the same  $rm$ , by the mutual distance measure.

#### 3.2 The Hart's Algorithm Modification

The Hart's algorithm can be used for the arranged original reference set according to  $mdm$  and next  $rm$ , or  $rm$  and next  $mdm$ . It is expected that the first points



**Fig. 1.** Representative and mutual distance measure for point  $x$  from circle class

from sorted reference set will be added to reduced reference set. The Hart’s algorithm adds the points from the beginning, but also the points from end of sorted reference set. However, the samples from the beginning are qualified more frequently than the ones from the end. The proposition of the present paper consists of breaking the cycle when new point is added to reduced reference set and starting the sample presentation from the beginning. The presented Hart’s algorithm modification consists of the use of the original reference set arranged by double sorting described above.

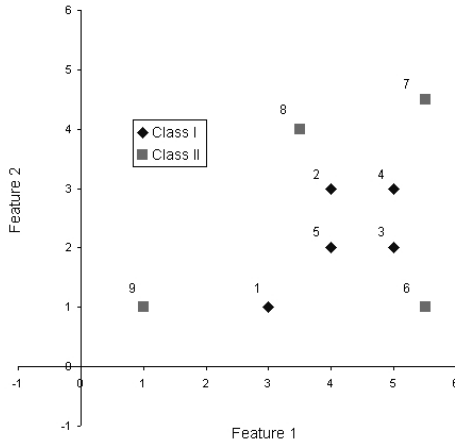
### 4 Simple Example

The Gowda-Krishna algorithm and the presented double sort algorithms will be illustrated by a simple example (tab. 1). The artificial reference set has nine two-dimensional points (fig. 2).

Table 2 presents sequence of points sorted increasingly by mutual distance measure (Gowda-Krishna algorithm), sequence of points sorted increasingly by mutual distance measure and then decreasingly by the representative measure ( $mdm-rm$  double sort algorithm) and sequence of points sorted decreasingly by

**Table 1.** Example artificial reference set

	point class	feature 1	feature 2
1	I	3,0	1,0
2	I	4,0	3,0
3	I	5,0	2,0
4	I	5,0	3,0
5	II	4,0	2,0
6	II	5,5	1,0
7	II	5,5	4,5
8	II	3,5	4,0
9	II	1,0	1,0



**Fig. 2.** Example artificial reference set in two-dimensional space

representative measure and increasingly by mutual distance measure (*rm-mdm* double sort algorithm).

The Gowda-Krishna arrangement promotes points, which lie near the class borders (almost all the points in the presented example), *rm-mdm* double sort algorithm - points, which the best represent its own class (point 5), and in the groups of equal representative measures - points, which lie near the class borders (in this example all the points with the same representative measure have also the same mutual distance measure) and *mdm-rm* double sort algorithm - inversely.

Table 3 shows the reduced reference sets produced by Hart’s algorithm from sorted reference sets.

**Table 2.** Points sorted by Gowda-Krishna and double sort algorithms (mdm - mutual distance measure, rm - representative measure)

Gowda-Krishna			<i>mdm-rm</i>				<i>rm-mdm</i>			
algorithm			double sort algorithm				double sort algorithm			
point	class	mdm	point	class	mdm	rm	point	class	rm	mdm
7	II	0	4	I	0	3	5	I	4	1
6	II	0	3	I	0	2	4	I	3	0
8	II	0	2	I	0	2	2	I	2	0
9	II	0	1	I	0	1	3	I	1	0
2	I	0	7	II	0	0	1	I	1	0
1	I	0	8	II	0	0	8	II	0	0
4	I	0	9	II	0	0	9	II	0	0
3	I	0	6	II	0	0	7	II	0	0
5	I	1	5	I	1	4	6	II	0	0

**Table 3.** Hart’s algorithm results for points from Gowda-Krishna sorting and double sorting

algorithm	reduced reference set
Gowda-Krishna, i.e. <i>mdm</i>	{1,2,3,6,7,8,9}
double sort ( <i>mdm-rm</i> )	{1,4,6,7,8,9}
double sort ( <i>rm-mdm</i> )	{5,6,8,9}

Gowda-Krishna algorithm reduces nine points of the reference set to seven points. The use of *rm-mdm* sorting and Hart’s procedure results only in four points and application of *mdm-rm* arrangement - six points.

The reduced reference set in Gowda-Krishna algorithm is built from points, which lie near class borders. Only two points were eliminated.

The *mdm-rm* double sort algorithm produced one point less than Gowda-Krishna algorithm. The point 4 is present in the reduced reference set instead of the points 2 and 3 as a consequence of different ways of sorting. Point 4 has the same mutual distance measure as points 2 and 3 but the higher representative measure. Therefore, it was added to the reduced reference set before points 2 and 3 analyzes.

The reduced reference set obtained by *rm-mdm* double sorting contains the point 5 from class I - the most representative point of this class, and the points 6, 8 and 9 from class II. Comparing this result with the result of Gowda-Krishna approach, it can be noticed that the point 5 was chosen instead of the points 1, 2 and 3. The point 7 was not added to reduced reference set, because it is correctly classified by point 8.

For the Gowda-Krishna arrangement, original Hart’s algorithm version was used and for double sort algorithms - the modified version. However in this simple example, it had not any influence on better results. The results of Hart’s original algorithm used for presented example and double sorting are the same.

## 5 Tests

The tests was made on two original sets:

- phoneme [5] - contains vowels coming from 1809 isolated syllables (three observation moments for each vowel have been kept). Five different attributes were chosen to characterize each vowel. Two different classes are distinguished, i.e. nasals (3818 patterns) and oral vowels (1586 patterns). The set was divided ten times into two sets of equal size: the training set and the validating set;
- satimage [5] - generated from Landsat Multi-Spectral Scanner image data. It contains 6435 patterns with 36 attributes (4 spectral bands x 9 pixels in neighborhood). The six following classes are distinguished: red soil (1533 patterns), cotton crop (703 patterns), grey soil (1358 patterns), damp grey

soil (626 patterns), soil with vegetation stubble (707 patterns) and very damp grey soil (1508 patterns). The set was divided ten times into the training and validating set of approximately equal size;

The three algorithms have been used for reduction the original reference sets:

- GK - the Gowda-Krishna arrangement (*mdm*) with Hart's original algorithm;
- DSA(*mdm-rm*) - *mdm-rm* double sort algorithm (main sorting by mutual distance measure) with Hart's algorithm modified version;
- DSA(*rm-mdm*) - *rm-mdm* double sort algorithm (main sorting by representative measure) with Hart's algorithm modified version;

The Euclidean metric and classic standardization (based on mean values and standard deviations of features) are used. The samples from the validation sets are classified by the use the 1-NN rule with the reduced reference sets. Experiments with the use the 1-NN rule and the complete reference set have been also performed. The classification error rates (in percentages), the number of points in reduced reference sets, the mean values and standard deviations from all ten measurements are presented in tables 4 and 5.

**Table 4.** Phoneme set results

sn	1NN GK		DSA( <i>rm-mdm</i> )		DSA( <i>mdm-rm</i> )		
	error r. [%]	error r. [%]	number of points	error r. [%]	number of points	error r. [%]	number of points
1	12,5	16,3	660	15,2	643	15,7	613
2	13,3	16,9	662	15,7	649	16,0	643
3	11,8	15,1	664	14,6	632	14,3	638
4	12,4	15,5	681	14,4	640	14,5	640
5	12,2	15,7	661	15,7	637	15,5	628
6	12,4	15,9	662	15,5	658	16,1	642
7	13,0	15,0	682	15,1	653	15,3	639
8	12,4	15,7	659	14,9	645	15,8	631
9	12,0	15,4	669	14,5	637	14,9	629
10	11,9	15,2	697	14,1	690	16,0	663
<b>mean v.:</b>	<b>12,4</b>	<b>15,7</b>	<b>669,7</b>	<b>15,0</b>	<b>648,4</b>	<b>15,4</b>	<b>636,6</b>
<b>st. deviat.:</b>	<b>0,45</b>	<b>0,55</b>	<b>12,08</b>	<b>0,54</b>	<b>15,76</b>	<b>0,61</b>	<b>12,27</b>

For phoneme test set, Gowda-Krishna algorithm average classification error rate increased by 26,6 % while in *rm-mdm* double sort algorithm - by 21 % and in *mdm-rm* double sort algorithm - by 24,2 % in comparison with classification using complete reference set (and 1-NN rule). The average size of the reduced reference set obtained by *mdm-rm* double sort algorithm is approximately 637 points while the reduced reference set obtained by Gowda-Krishna algorithm - ca. 670 points. Overall, for phoneme set, the *rm-mdm* double sort algorithm results in the best classification and the *mdm-rm* double sort algorithm - in the smallest size of consistent reduced reference set.

**Table 5.** Satimage set results

sn	1NN	GK	DSA( <i>rm-mdm</i> )		DSA( <i>mdm-rm</i> )		
	error r. [%]	error r. [%]	number of points	error r. [%]	number of points	error r. [%]	number of points
1	10,7	16,3	660	13,1	600	13,9	581
2	11,3	16,9	662	13,8	570	13,6	603
3	11,0	15,1	664	12,3	587	12,9	588
4	10,4	15,5	681	12,0	621	12,3	629
5	10,7	12,9	641	13,1	573	12,9	577
6	10,0	12,8	644	12,6	617	13,2	614
7	10,5	11,9	678	12,2	612	12,3	619
8	10,3	12,9	635	13,0	598	12,9	588
9	11,0	13,6	621	13,0	580	12,7	590
10	10,3	12,6	642	12,6	595	13,9	606
<b>mean v.:</b>	<b>10,6</b>	<b>14,1</b>	<b>652,8</b>	<b>12,8</b>	<b>595,3</b>	<b>13,1</b>	<b>599,5</b>
st. deviat.:	<b>0,37</b>	<b>1,65</b>	<b>18,31</b>	<b>0,51</b>	<b>16,97</b>	<b>0,57</b>	<b>16,49</b>

For satimage test set, the mean misclassification rate of Gowda-Krishna algorithm increased by 33 % while in *rm-mdm* double sort algorithm - increased only by 20,8 % and in *mdm-rm* double sort algorithm - by 23,6 % as compared to classification using the 1-NN rule with the complete reference set. The reduced reference sets obtained by double sort algorithms contained average only approximately 597 points while the reduced reference set received by Gowda-Krishna algorithm - ca. 653 points. Overall, for satimage set, the double sort algorithms result in the better classification quality and in the smaller size of consistent reduced reference set than Gowda-Krishna modification of Hart’s procedure.

## 6 Conclusions

The Hart’s algorithm was the first proposal for reference set reduction. It was improved by Gowda and Krishna. Their modification consisted of the preliminary arrangement of the original reference set on the basis of the mutual distance measure. In the approaches proposed in this paper an additional key for preliminary sorting the primary reference set was applied. Thus, the original reference set was sorted by two keys: mutual distance measure and representative measure, which were used in two possible different orders. Apart from the new proposition of preliminary sorting the original reference set, the modification of Hart’s procedure was introduced. It consists of different presentation of samples during their qualification to the reduced reference set. In contradiction to the Hart’s algorithm and its modification done by Gowda and Krishna, the presentation of the sequential points is breaking and starts from the beginning.

The reference set reduction algorithms using double point sorting result in the better classification quality and smaller size of reduced reference set in comparison with Gowda-Krishna modification.



## References

1. Hart P.E. (1968) The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, vol. IT-14, no. 3 (Corresp.), pp. 515-516
2. Gowda K. C. and Krishna G. (1979) The condensed nearest neighbor rule using the concept of mutual nearest neighborhood. *IEEE Transaction on Information Theory*, v. IT-25, no. 4, pp. 488-490
3. Theodoridis S., Koutroumbas K. (2006) *Pattern Recognition - Third Edition*. Academic Press - Elsevier, USA
4. Duda R.O., Hart P.E., Stork D.G. (2001) *Pattern Classification - Second Edition*. John Wiley & Sons, Inc
5. Newman D.J., Hettich S., Blake C.L., Merz C.J. (1998), *UCI Repository of machine learning databases*. Irvine, CA: University of California, Department of Information and Computer Science [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]

---

# Using Graph Mining Methods in Searching for Frequent Patterns in Graph-Based Design Representation

Barbara Strug

Faculty of Physics, Astronomy and Applied Computer Science,  
Jagiellonian University, Reymonta 4, Cracow, Poland  
uistrug@cyf-kr.edu.pl

**Summary.** This paper presents an approach based on graph mining techniques to searching for frequent patterns in design systems. The motivation for such approach is briefly presented. The graphs-based representation is used as an underlying structure for design representation. Different types of graphs used to represent designs are presented. Especially hierarchical labelled graphs are described. A possibility of adapting graph mining methods for design problems is also shown.

## 1 Introduction

Graphs are very useful as a mean to represent complex objects in different domains of computer science [22]. Their ability to represent the structure of an object as well as the relations of different types between its components makes them particularly useful in computer aided design. They can represent an artifact being designed taking into account the inter-related structure of many design objects i.e. the fact that parts of an object can be related to other parts in different ways. As designing new artifacts requires a method of generating representing them graphs many methods for graph generation were researched.

These methods include approach based on application of the theory of formal languages to the computer aided design [21, 8, 9, 11], in particular the graph based representation jointly with graph grammars [2, 8, 9, 10, 12, 20], and grammar systems [3, 6, 7, 23, 19, 16]. Other methods used to generate graphs representing designs include evolutionary computations that were used in different domains of design [2, 12, 20].

All these generation methods result in producing a number of graphs representing designs. Using graph grammars and evolutionary methods, either separately or combined together, we can build large database of graphs - or designs.

The main problem lies in the complexity and size (understood as the number of graphs) of such database. It is difficult to automatically evaluate the quality of these graphs. The quality of graph is here understood as the quality of the design a graph represents as a solutions to a given design problem. Thus the process of evaluation usually requires the presence of a human designer who can choose the best solution or give some numerical values to each design. The

problem is that for a human "evaluator" to work all graphs has to be rendered to designs. This process in turn requires visualising designed objects. Until we have a very simple design problem the rendering of designs may be very complex and result in very long execution times. This problem is especially important in evolutionary design systems where large numbers of graphs are generated in each population and a speed of fitness evaluation is a key factor in getting a usable design system. In such situations it would be useful to be able to evaluate graphs on the basis of the evaluation of graphs in earlier populations. The simplest and most appropriate method seems to be one using a number of graphs representing designs for which a human "evaluator" has defined a quality value as a basis for evaluating other designs in the same design problem. As it can be noticed that the designs getting higher quality values usually have some common elements, finding frequent substructures in graphs seems a useful approach.

Thus the use of methods based on frequent graph mining techniques is proposed in this paper. Two main types of such methods are considered here: the a priori based methods and the ones based on pattern growth algorithm. Some problems that arise with the use of such methods are also briefly presented.

## 2 Graph-Based Design Representation

The methods used in CAD problems like boundary representations, sweep-volume representation, surface representations or CSG (constructive solid geometry) [17, 14, 18] allow only for the "coding" of geometry of an object being designed and do not take into account the inter-related structure of many design objects i.e. the fact that parts (or components) of an object can be related to other parts in different ways. In this paper a representations based on hierarchically organized data is used.

Different types of graphs have been researched and used in this domain, for example composition graphs [11, 10]. In this paper an extension of composition graphs - hierarchical graphs is used. They can represent an artifact being designed at different levels of detail at different stages of the design process thus hiding unnecessary low-level data and presenting the designer only an outline of the object or showing a detailed view of the whole object (or of its part).

Hierarchical graphs (HGs) are an extension of traditional graphs. They consist of nodes and edges. What makes them different from standard graphs is that nodes in HGs can contain internal nodes. These nodes, called children, can in turn contain other internal nodes and can be connected to any other nodes with only exception being their ancestors.

A node in a hierarchical graph may represent a geometrical object or a groups of objects. It may also be used to hide certain details of a designed object that are not needed at a given stage of design or to group object having some common features (geometrical or functional). A node that represents a single object is called an object node. Nodes that do not represent actual geometric entities but are used to represent hierarchical structure or other relations are called group nodes. It is important to note that the edges between nodes are by no means

limited to edges between descendants of the same node. In other words there may exist edges between nodes having different ancestors.

Nodes and edges in hierarchical graphs can be labelled and attributed. Labels are assigned to nodes and edges by means of node and edge labelling functions respectively, and attributes - by node and edge attributing functions. Attributes represent properties (for example size, position, colour or material) of a component represented by a given node. It should be noted here that a single node may be referred to by its own label or by a so called *full label* which contains labels of its all ancestors in the hierarchical structures separated by a dot. For example a node labelled  $x$  which is a child node of atop level node labelled  $y$ , may be referred to either as  $x$  or  $y.x$ .

A labelled attributed hierarchical graph may represent a potentially infinite number of designs. The given graph  $G$  being a hierarchical graph can represent a, potentially, infinite subset of designs having the same structure. To represent an actual design we must define an instance of a graph. An instance of a hierarchical graph is a hierarchical labelled attributed graph in which to each attribute a value has been assigned from the set of possible values of a given attribute.

As such a hierarchical graph defines only a structure of a design to create a visualisation of an object an interpretation is necessary. Such interpretation determines the assignments of geometrical objects to nodes, its fragments to bonds and establishes a correspondence between edges and sets of relations between objects (components of a design). The objects assigned to nodes are usually called primitives.

### 3 Frequent Graph Mining Methods

Frequent graph mining techniques are developed on the basis of a more general frequent pattern mining. Frequent patterns can be defined as itemsets, subsequences, or substructures which appear in a data set with a frequency equal or greater to a user-specified threshold value. In this paper the most interesting are the substructures patterns. A substructure can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences. If a substructure occurs frequently in a graph database, it is called a (frequent) structural pattern. Finding frequent patterns is an important part in mining associations, correlations, and many other interesting relationships among data. It may also be used in data indexing, classification, clustering, and other data mining tasks.

Frequent pattern mining was first proposed by Agrawal et al. [1] for market basket analysis in the form of association rule mining. It was used to analyze customer buying habits by finding associations between the different items that customers place in their "shopping baskets". Since the first proposal of this data mining problem and the mining algorithms relating to it, there have been many other research publications, on various kinds of extensions and applications, from scalable data mining methodologies, to handling a wide diversity of data types and a huge number of applications [25, 24].

### 3.1 Basic Notions

Let a  $DG = \{G_1, G_2, \dots, G_n\}$  be a set of graphs, then  $support(g)$  or  $frequency(g)$  is a percentage (or number) of graphs in  $DG$  of which  $g$  is a subgraph. A graph  $g$  is frequent if its support is no less than a minimum threshold value  $MinSupport$ .

### 3.2 Apriori-Based Methods

The basic frequent subgraph mining algorithm ([1]) is presented in fig. 1. The frequent patterns of larger size are searched for starting from subgraphs of smaller size by adding to them an additional node or edge with a node.

#### **Apriori(DG, MinSupport, $SG_k$ )**

Input:  $DG$  - set of graphs

$MinSupport$  - a minimum frequency threshold value

Output:  $SG_k$  - set of frequent graph of size  $k$

**begin**

$SG_{k+1} := \emptyset$

**foreach** frequent  $g_i \in SG_k$  **do**

**foreach** frequent  $g_j \in SG_k$  **do**

**foreach**  $g$  of size  $(k + 1)$  generated by the merging of  $g_i$  and  $g_j$  **do**

**if**  $g$  is frequent in  $DG$  and  $g \notin SG_{k+1}$  **then** insert( $g, SG_{k+1}$ )

**if**  $SG_{k+1} \neq \emptyset$  **then** Apriori(DG, MinSupport,  $SG_{k+1}$ )

**end**

**Fig. 1.** An outline of Apriori based algorithm

The size of graph is understood in this paper as a number of nodes of the graph at the top level of hierarchy (without counting the nodes grouped by a given node). The main reason for complexity of the Apriori-based algorithm is a candidate solution generation step. Various types of candidate solution generation algorithms were proposed. One of them is so called AGM algorithm proposed by Inokochi [15], which increases the substructure by one node at each iteration of algorithm 1. Two frequent graphs of size  $k$  are combined only if they have the same subgraph of size  $k - 1$ . The newly generated graph of size  $k + 1$  includes the common subgraph of size  $k - 1$  and two additional nodes from both size  $k$  graphs. As it is undetermined if there is an edge connecting the two nodes from combined graphs two different  $k + 1$  size graphs are possible.

This algorithm is a basic formulation for general graphs. In case of using hierarchical graphs some modifications must be introduced. Firstly, as the size of a hierarchical graph is understood in this paper as the number of top level nodes, to find a common subgraph only top level nodes are taken into account. So actually the graphs can be of different size yet be considered of size  $k$ . Secondly, the two additional nodes are joined by a new edge if it does not lead to introducing a child-parent edge, which is prohibited in a hierarchical graph definition used

here. It removes the undeterminism present in the original algorithm. Moreover, if a node is added to a graph size  $k$  and this new node is a hierarchical one all its descendants are also added to a new graph.

It can be noticed that Apriori-based algorithms require a lot of computations to join two size  $k$  graphs into one size  $k + 1$  graph candidate solution. To reduce these computations other methods were developed, mainly based on pattern growth model.

### 3.3 Pattern Growth Methods

Pattern growth methodology consists in extending a single pattern into more complex one [13]. A single graph  $g$  can be extended by an edge  $e$ . This edge may or may not introduce a new node into graph. This extension of a graph  $g$  is denoted by  $ext(g, e)$ . Algorithm shown in fig.2 shows a basic framework for this approach. For each found frequent graph  $g$  it tries recursively to extend it until all possible frequent graphs of which  $g$  is a subgraph are found. The recursive process ends when no more frequent graph can be found.

The main problem of this method is the fact that each  $k$  size frequent graph can be found independently  $k$  times - by extending  $k$  different size  $k - 1$  graphs. While it is a problem for general graph mining task, it is less likely to happen in design problems. Moreover using hierarchical graphs strongly reduces the number of edges that can be used to extend a subgraph, as only edges connecting nodes on the same level of hierarchy are allowed.

#### PatternGrowth( $g, DG, MinSupport, SG$ )

Input:  $g$  - a frequent graph

$DG$  - set of graphs

$MinSupport$  - a minimum frequency threshold value

Output:  $SG$  - set of frequent graph

**begin**

**if**  $g \in SG$  **then** exit()

**else** insert( $g, SG$ )

scan  $DG$  to find all edges  $e$  such then  $g$  can be extended by  $e$  to  $ext(g, e)$

**foreach** frequent  $ext(g, e)$  **do**

    PatternGrowth( $ext(g, e), DG, MinSupport, SG$ )

**end**

**Fig. 2.** An outline of pattern growth algorithm

## 4 Frequent Design Pattern Mining

Both algorithms described in the previous section were tested on a chair design problem. As the basis for testing two small sets of graphs were used. The graphs were generated by two methods: graph grammar based approach and an

evolutionary algorithm. The design problem used is rather simple, but very useful in testing different algorithms. The graphs representing chairs have usually no more than 50 nodes. The graph set generated by a graph grammar consisted of 100 elements and the one produced by an evolutionary algorithm also of 100 (taken from different populations). In both cases two methods of nodes matching were used. In the first one two nodes were considered identical if their labels were equal; in the second one the equality of the full label was required to match nodes.

In the first test the Apriori-based algorithm adapted to hierarchical graphs was used on both test graph sets. The resulting set of frequent sungraphs was computed relatively fast for the graph grammar generated set. For the set generated by the evolutionary algorithm it took much longer to compute the frequent graph set. One of the reasons for this difference may be the relative similarity of graphs generated by the same grammar. There is a finite and limited number of productions in a grammar so the number of different graphs substructures introduced by these productions is also limited. Thus the step of joining two size  $k$  graphs into one size  $k + 1$  which requires finding their common core of size  $k - 1$  is much faster. On the other hand the operators used in evolutionary computations introduce much more diversity into graph set. But to fully verify this possibility more testing, possibly on much larger graph sets, is needed.

In the case of pattern growth approach the set of frequent graphs was found slightly faster of the graphs generated by evolutionary method but the time difference is too small to allow for any conclusion and it will also need more testing.

## 5 Concluding Remarks and Future Work

In this paper two methods of finding frequent subgraphs in hierarchical graphs used to represent designs are presented and adapted to hierarchical graphs. This methods were tested on a small database of graphs generated by different methods (graph grammars and evolutionary algorithm). This preliminary test seems to indicate that the apriori-based algorithm is faster on graph sets generated by a graph grammars, while the pattern growth method is better in finding frequent substructures in a population generated by an evolutionary algorithm. It seems that the cause of this result lies in the fact that the graphs generated by a graph grammar show significantly less diversity than the ones generated by evolutionary method, but there is need for much more testing to verify it.

In the further exploration of the problem it is planned to test both methods on a more complex problems (larger graph sets and more difficult design problem) to determine whether there is a real difference of performance of the apriori-based algorithm and pattern growth one.

In this pattern an exact matching of graphs was used i.e. the set of graphs is scanned and a pattern is considered frequent only if it is a subgraph of a high enough number of graphs in the set. Yet in many situations, and in design practice especially, it would be more useful to find a pattern that occurs in many graphs but in a slightly different form. To be able to find such patterns a

good similarity measure for graphs is needed. It could then be used to search for similar subgraphs.

Another problem for future work is adapting the frequent graph mining algorithms to another design representation - hypergraphs. Due to their ability to represent multiple-argument relations, as opposed to binary relations in traditional graphs, hypergraphs are more and more often used in design systems. So finding an algorithm searching for frequent hypergraphs could be very useful.

## References

1. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proc. 1993 ACM-SIGMOD int. conf. management of data (SIGMOD'93), Washington, DC, pp 207-216
2. Borkowski A., Grabska E., Nikodem P, and Strug B. Searching for Innovative Structural Layouts by Means of Graph Grammars and Evolutionary Optimization, Proc. 2nd Int. Structural Eng. And Constr. Conf, Rome 2003.
3. E. Csuhaj-Varjú, J. Dassow, J. Kelemen and Gh. Paun. Grammar systems. A grammatical approach to distribution and cooperation. Topics in Computer Mathematics 8. Gordon and Breach Science Publishers, Yverdon, 1994.
4. E. Csuhaj-Varjú, J. Dassow, and Gh. Paun. Dynamically controlled cooperating/distributed grammar systems. Information Sciences, 69(1-2), pp. 1-25, 1993.
5. E. Csuhaj-Varjú and Gy. Vaszil. On context-free parallel communicating grammar systems: Synchronization, communication, and normal forms. Theoretical Computer Science, 255(1-2), pp. 511-538, 2001.
6. E. Csuhaj-Varjú: Grammar systems: A short survey, Proceedings of Grammar Systems Week 2004, 141-157, Budapest, Hungary, July 5-9, 2004.
7. J. Dassow, Gh. Paun, and G. Rozenberg. Grammar systems. In A. Salomaa and G. Rozenberg, editors, Handbook of Formal Languages, volume 2, chapter 4, pp. 155-213, Springer-Verlag, Berlin-Heidelberg, 1997
8. E.Grabska. Theoretical Concepts of Graphical Modelling. Part one: Realization of CP-graphs. Machine GRAPHICS and VISION, 2(1), pp. 3-38, 1993.
9. E.Grabska, Theoretical Concepts of Graphical Modelling. Part two: CP-graph Grammars and Languages. Machine GRAPHICS and VISION, 2(2),pp. 149-178, 1993.
10. E.Grabska, W. Palacz. Hierarchical graphs in creative design. MG&V, 9(1/2), pp. 115-123, 2000.
11. E. Grabska. Graphs and designing. Lecture Notes in Computer Science, 776 (1994).
12. E. Grabska, P. Nikodem, B. Strug. Evolutionary Methods and Graph Grammars in Design and Optimization of Skeletal Structures Weimar, 11th International Workshop on Intelligent Computing in Engineering, Weimar, 2004.
13. J. Han, J. Pei, Y. Yin, and R. Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-pattern Tree Approach. Data Mining and Knowledge Discovery: An International Journal, Volume 8, Issue 1, pages 53-87, January 2004, Kluwer Academic Publishers.
14. Hoffman, C. M., Geometric and Solid Modeling: An Introduction, Morgan Kaufmann, San Francisco, CA, 1989
15. A. Inokuchi, T. Washio, H. Motoda, An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data, Principles of Data Mining and Knowledge Discovery: 4th European Conference, PKDD 2000, Lyon, France, September 2000. Proceedings, pp.87-92



16. J. Kelemen. Syntactical models of cooperating/distributed problem solving. *Journal of Experimental and Theoretical AI*, 3(1), 1-10, pp. 1991.
17. Mantyla, M., *An Introduction To Solid Modeling*, Computer Science Press, Rockville, MD, vol. 87, 1988
18. Martin, R R and Stephenson, P C Sweeping of Three-dimensional Objects. *Computer Aided Design* Vol 22(4), 1990, pp 223-234.
19. C. Martín-Vide and V. Mitrana. Cooperation in contextual grammars. In A. Kelemenová, editor, *Proceedings of the MFCS'98 Satellite Workshop on Grammar Systems*, pp. 289-302. Silesian University, Opava, 1998.
20. P. Nikodem and B. Strug. *Graph Transformations in Evolutionary Design*, Lecture Notes in Computer Science, vol 3070, pp. 456-461, Springer, 2004.
21. Rozenberg, G. *Handbook of Graph Grammars and Computing by Graph. Transformations*, vol.1 Foundations, World Scientific London 1997.
22. Rozenberg, G. *Handbook of Graph Grammars and Computing by Graph. Transformations*, vol.2 Applications, Languages and Tools, World Scientific London 1999
23. M. Simeoni and M. Staniszkis. Cooperating graph grammar systems. In Gh. Paun and A. Salomaa, editors, *Grammatical models of multi-agent systems*, pp. 193-217. Gordon and Breach, Amsterdam, 1999.
24. X. Yan, P. S. Yu, and J. Han, Substructure Similarity Search in Graph Databases, *SIGMOD'05 (Proc. of 2005 Int. Conf. on Management of Data)*,
25. X. Yan, P. S. Yu, and J. Han, Graph Indexing: A Frequent Structure-based Approach, *SIGMOD'04 (Proc. of 2004 Int. Conf. on Management of Data)*, 2004

---

# Stability of Feature Points Defined by Approximating Quadric Surfaces

Leszek Luchowski

Institute of Theoretical and Applied Computer Science  
leszek.luchowski@iitis.gliwice.pl

**Summary.** The analysis of freeform surfaces requires them to be reduced to more regular geometric structures. One approach is to approximate fragments of the given surface by quadrics, then use the points, lines, and planes defined by the quadric as bases for measurements. The present work describes a series of experiments designed to assess the stability of points defined in this way on human facial features. Both positive and negative findings are discussed.

## 1 Introduction

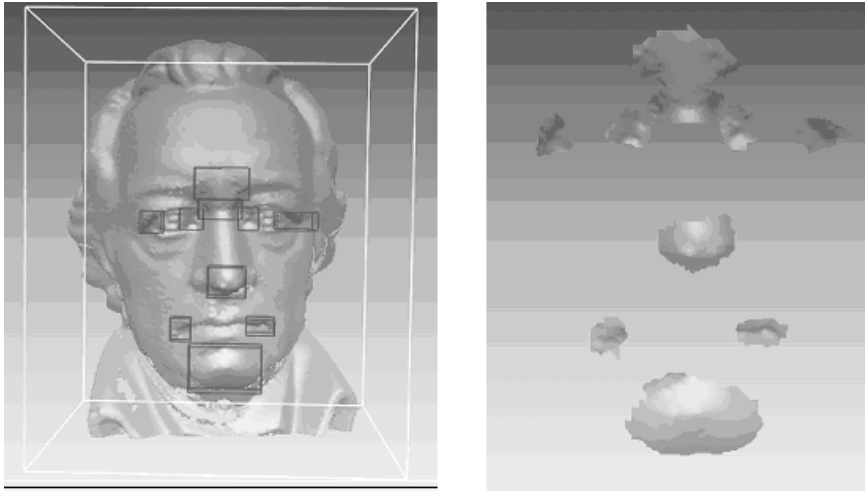
When applying computer vision to biological and, especially, medical objects, we are often confronted with the need to define visual features which have previously only been identified by human perception.

In particular, to measure distances and angles on freeform surfaces, we first have to construct geometric primitives such as points (or lines, or planes) that will define our measurements. The primitives should depend stably on surface data. In the case of polygon-tesselated surfaces, this means a primitive should be derived from a large number of mesh vertices or facets while avoiding noise-sensitive operations such as derivation or maximum search.

In the present work we shall concentrate on approximating a group of mesh vertices by a quadric surface, then taking the origins of the quadrics (i.e. the origin of the coordinate system in which the quadric has a canonical form) or their other identifiable points (e.g. saddle points, apices $\tilde{E}$ ), as our point primitive. As quadric approximation uses statistical moments from a population of points which can be arbitrarily large, it meets our requirement that primitives not depend strongly on single mesh vertices.

Existing definitions of anatomical measurement points are usually suited for human interpretation, using highly non-linear terms such as "deepest point on the curvature at the base of the nose" [1]. "Deepest" here suggests an extreme of a coordinate, but no coordinate is defined. In this work, we try to interpret and implement these definitions, but in terms of the approximating quadrics, rather than of the raw 3D surfaces.

Our purpose is twofold: first, to obtain a repeatable, stable way of refining measurement points roughly indicated by a human operator. This should significantly reduce both work load and subjective error. The second and ultimate goal is to locate such points automatically. While achieving this will require much



**Fig. 1.** Left: scanned 3D image of the bust of Goethe with frames showing selected feature areas. Right: manually cropped feature areas.

further research, the present work examines the precision of what may be the final processing stage of such segmentation.

We shall now examine the stability of quadric-defined points under the variation of such factors as instance of scanning and choice of vertex group.

## 2 Experiment

We used two dummy faces, to avoid shape changes due to breathing and facial expression. The actual models were the head of a doll (one with natural-looking features) and a bust of J.W.Goethe. Ten feature areas were chosen:

- GB - glabella: the most prominent point on the skin above the nose
- En - endocanthion: the inner corner of the eye fissure (left and right)
- Ex - exocanthion: the outer corner of the eye fissure (left and right)
- Hn - cutaneous nasion: the deepest point at the base of the nose
- Ap - nasal apex: the tip of the nose
- Ch - cheilion: corner of mouth (left and right)
- Pg - Soft tissue pogonion: Most anterior point on chin in midsagittal plane

Either head was scanned three times with the Minolta Vi-9i laser scanner, from slightly different viewpoints. The resulting 6 images (3D shells) were manually cropped to leave only small areas surrounding the points in question (Fig. 1). Cropping was also performed three times, under subjective variation.

The cropping resulted in 180 small 3D images, each consisting of one surface patch representing a particular feature area of a specific scan of one of the objects

(Goethe bust or doll). The patches were approximated by quadric surfaces, and the quadric polynomials used to define characteristic points.

Almost all of the features were classified into 3 quadric categories: ellipsoids, and one- and two-sheet hyperboloids (Table 1). Only one instance of cheilion was classified as a cone. Other quadric types were not represented.

**Table 1.** Quadric types. EL: elipsoid, 1H/2H: 1- or 2-sheet hyperboloid, CN: cone.

Model	Goethe bust									Doll								
	1			2			3			1			2			3		
Scan	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Crop	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Ap	EL	EL	EL	EL	EL	EL	EL	EL	EL	EL	EL	EL	EL	EL	EL	EL	EL	EL
Gla	2H	2H	2H	2H	1H	1H	EL	2H	2H	EL	2H	2H	EL	EL	1H	2H	2H	2H
Po	EL	EL	EL	EL	EL	EL	EL	EL	EL	2H	2H	2H	2H	2H	EL	2H	EL	EL
CheR	1H	1H	1H	1H	1H	1H	1H	1H	1H	EL	EL	1H	1H	EL	1H	EL	EL	EL
CheL	2H	CN	2H	1H	2H	2H	1H	EL	2H	1H	EL	EL	1H	1H	2H	EL	EL	EL
Na	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H
EnCR	2H	2H	2H	2H	2H	1H	EL	1H	1H	2H	2H	2H	1H	2H	2H	2H	2H	2H
EnCL	2H	2H	EL	EL	2H	2H	1H	2H	2H	2H	2H	2H	2H	2H	2H	2H	2H	2H
ExCR	1H	2H	1H	1H	1H	2H	2H	2H	2H	1H	EL	1H	1H	1H	1H	1H	1H	1H
ExCL	1H	2H	1H	1H	1H	1H	2H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H	1H

The final result of this processing, for each scan/cropping combination, was a set of points in 3D space. The relative position of points was compared between sets in order to assess the stability of quadric-based points with respect to changing scanning conditions and subjective cropping.

**Table 2.** Feature dispersion and approximation error

Feature	Goethe		Doll	
	Av.D	QE	Av.D	QE
Ap	1.09	0.01	5.06	0.03
Gla	4.48	1.61	10.13	0.02
Po	3.48	0.03	2.20	0.03
CheR	3.09	3.21	3.83	0.21
CheL	7.76	8.76	4.30	0.52
Na	1.64	0.04	1.83	0.02
EnCR	4.09	1.26	1.44	0.06
EnCL	5.09	16.52	0.62	0.07
ExCR	3.81	8.29	6.53	4.53
ExCL	3.63	313.56	4.53	832.82

### 3 Results

Absolute feature positions can only be compared within each scan (different scans have different coordinate systems). For each scan, three positions of each feature were determined, using the three different manual croppings. The distances between such positions within each scan, averaged over the three distances per scan and over the three scans, were computed for each feature as a measure of repeatability; these average distances, in millimeters, are presented in Table 2,

**Table 3.** Avg standard deviations of feature-to-feature distances in Goethe bust...

	Ap	GB	Pg	Ch(R)	Ch(L)	Hn	En(R)	En(L)	Ex(R)	Ex(L)
GB	4,30	0,00								
Pg	2,87	4,49	0,00							
Ch(R)	2,60	3,86	1,99	0,00						
Ch(L)	3,77	7,35	5,12	2,22	0,00					
Hn	1,54	4,52	1,69	1,70	4,46	0,00				
En(R)	1,50	2,45	3,03	2,16	4,39	2,00	0,00			
En(L)	1,58	6,43	3,11	1,91	5,66	1,92	2,06	0,00		
Ex(R)	3,61	2,78	3,09	3,68	3,92	3,65	3,78	4,09	0,00	
Ex(L)	3,36	7,56	2,43	2,01	5,98	2,42	2,11	2,90	3,79	0,00

... and in doll face:

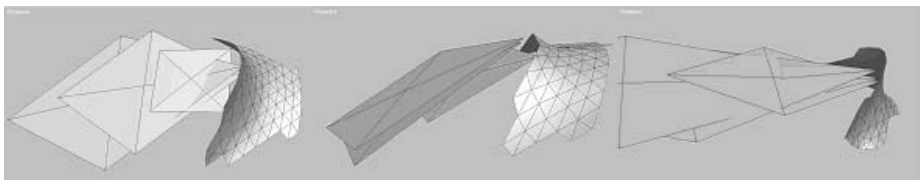
	Ap	GB	Pg	Ch(R)	Ch(L)	Hn	En(R)	En(L)	Ex(R)	Ex(L)
GB	3,76	0,00								
Pg	2,28	3,80	0,00							
Ch(R)	2,47	2,67	1,57	0,00						
Ch(L)	2,12	4,21	1,29	1,62	0,00					
Hn	1,00	3,93	1,00	1,01	1,33	0,00				
En(R)	1,07	1,83	1,28	1,03	1,44	1,18	0,00			
En(L)	1,37	5,36	0,86	0,97	0,73	0,74	0,70	0,00		
Ex(R)	3,30	2,21	2,68	3,33	2,98	3,33	3,22	2,95	0,00	
Ex(L)	3,05	7,29	0,90	1,59	2,03	1,97	1,48	1,78	3,53	0,00

column "Av.D". The QE columns represent the approximation error in  $mm^2$  per vertex in the patch.

### 4 Positive Findings

Certain feature areas have proven to yield very stable approximating quadrics. One example is the nasion of the doll (Fig. 2). In this case, the approximating one-sheet hyperboloid determined the inflexion point with a repeatability of less than 2mm, even when it was outside the cropped area (Fig. 2 center).

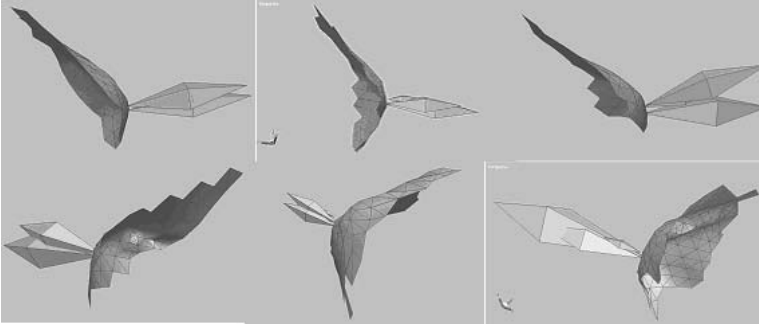
Overall, it can be said that when a quadric does not allow a local feature point to be identified reliably, this situation is indicated by a large approximation error.



**Fig. 2.** The nasion of the doll face; the three pictures represent three separate scanings, and the three kite-shaped octahedra in each picture point to feature points derived from three different manual clippings of each scan

Table 2 shows that a repeatability of less than 2mm is associated with quadric approximation errors of less than 0.1 ( $mm^2$  per vertex).

The two endocanthions of the doll face were approximated by two-shell hyperboloids in all but sample, and Table 2 confirms that they were the most reliably localized points in the experiment. In Figure 3 the cigar-shaped octahedra indicate the feature points detected (the tip of the octahedron closest to the surface is the relevant point; the opposite one represents the apex of the other shell of the hyperboloid, which has no physical meaning).



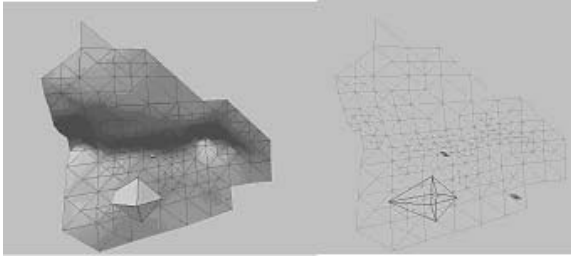
**Fig. 3.** Top: left endocanthion of the doll face; bottom: its right endocanthion. Three pictures of each represent separate scans. Three oblong octahedra in each picture (some occluded) pinpoint locations derived from three different manual croppings.

## 5 Negative Findings

Some of the selected areas were not consistently represented by the same type of quadric across instances of scanning and cropping. This makes the points incomparable and essentially invalidates this part of the data. It also led to serious discrepancies in the location of resulting points. The worst case was the left cheilion (corner of mouth) of the Goethe bust (Fig. 4). While a corner of a human mouth can arguably be represented by the saddle point of a hyperbolic paraboloid, this assumption would only hold for a very small area, and consequently for a small subset of mesh vertices in our scanned images. The larger areas around the cheilion, as cropped by hand for our experiments, could not be meaningfully approximated by a second-order surface.

## 6 Attempts at Automatic Segmentation

To identify salient features on a freeform 3D surface, we sought fragments that can consistently be represented by quadrics. This part of our experiment used a live human face, as we were not testing repeatability at this stage.



**Fig. 4.** One scan (third) of the left cheilion of the Goethe bust, with three different feature points determined by three different approximating quadrics. Right: the same model displayed in transparent mode to show the two minuscule octahedra. For the other scans, results were even more incoherent, and difficult to illustrate intelligibly.

The program created as many initial patches of a preset topological radius as would fit into the mesh. Then the patches evolved under five rules:

1. Expansion. A patch examines adjoining unassigned vertices to see if any of them fits the quadric currently approximating the patch. If the fit is better than a preset tolerance, the vertex can be added to the patch.
2. Competition. A patch can conquer an adjoining vertex from another patch, if the vertex fits the quadric of the former better than that of the latter.
3. Drift. A patch can "slide off" a border vertex (= one with neighbours outside the patch) if it fits its quadric less well than a predefined threshold.
4. Decay. If a patch has become too small to provide enough support for a meaningful quadric, it is eliminated and its vertices become unassigned.
5. Seeding. If sufficiently many contiguous unassigned vertices are found, a new patch is formed by the same process as created the initial patches.

The technique worked flawlessly for simple solids (cylinders, cubes etc). For a human face, setting tolerances became difficult. When tuned for looser approximations, the procedure would split the entire face into very few large ellipsoid fragments (one or two of which represented the cap over patients' hair), except for the nose which was then the only detected feature. With tighter tolerance, large uniform areas such as cheeks would split unnaturally.

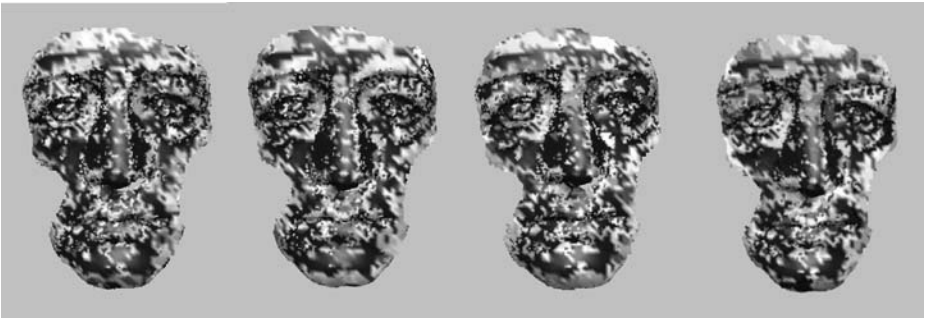
More details and results can be found in [6], from where Fig. 5 is quoted:



**Fig. 5.** Left to right: input mesh, initial patches, iterations 129, 300, and 999

## 7 Seeking Convergence Areas

Our next experiment consisted in using each vertex of the model in turn as a seed, to start a single quadric patch of limited size and let it evolve until it reaches stability. The resulting quadric then determined a feature point (the pole of the quadric closest to the seed). After all the vertices in the model had been used in this way, distances were measured between the feature points they yielded, and feature points no further apart than  $R$  were labeled as a group (the experiment was repeated for  $R=2, 3, 4$  and  $5\text{mm}$ ). All seed vertices whose resulting feature points were in one group were then marked with the same colour in Figure 6. Points which did not classify into a group were colored black. The color layout confirms that certain areas yield stable quadric approximations in the sense that the feature point defined by the quadric only weakly depends on the initial point used to start the evolution of the quadric patch. Such areas include the endocanthions, the nasion, and the nasal apex, which concurs with the findings discussed in Section 4.



**Fig. 6.** Vertices of the Goethe bust classified into equivalence groups for  $R=2, 3, 4$  and  $5\text{mm}$  (left to right)

## 8 Conclusions

Quadrics have the potential to be valuable for feature localization. They are the simplest polynomials representing all the basic local properties of a surface and extracting its characteristic points; they can also define other geometric references such as lines and planes. However, no single general-purpose algorithm is likely to detect all the different features of a human face; to achieve this, a priori knowledge obviously must be used.

**Acknowledgement.** This work was supported by the Ministry of Science and Higher Education of the Polish Government, under the research project No. 3 T11C 028 27.



## References

1. P.Diedrich, ed.: Ortodoncja. Rozwoj struktur ustno-twarzowych i diagnostyka (Orthodontics. Development and diagnosis of bucco-facial structures). In Polish. Wydawnictwo medyczne Urban i Partner, ( Wroclaw 2004)
2. Kowalski P., Tomaka A., Lisniewska-Machorowska B., Landmarks identification using active appearance models. *Archiwum Informatyki Teoretycznej i Stosowanej*, t. 17, , z 4/2005, pp 251-264
3. Chaari A., Luchowski L., Rousseau J., Multiple-view stereo matching of points using epipolar constraints and relaxation, *Archiwum Informatyki Teoretycznej i Stosowanej*, Vol. 17, No 4, 2005, pp. 241-250
4. Tomaka A., Lisniewska-Machorowska B., The application of the 3D surface scanning In the facial features analysis, *Journal of Medical Informatics and Technologies*, Vol. 9, October 2005, pp. 233-240
5. Tomaka A., Luchowski L., Skabek K., Tarnawski M., 3D Head surface scanning techniques for orthodontics, *Journal of Medical Informatics and Technologies*, Vol. 9, October 2005, pp. 123-130
6. Luchowski L., Segmenting 3d mesh images of the human face by local quadric parameterization, MIT 2006 Medical Informatics and Technology, Edited by: Ewa Pietka, Jacek Leski, Stanislaw Franiel, Wisla, 2006

---

# Generalisation of a Language Sample for Grammatical Inference of GDPLL( $k$ ) Grammars

Janusz Jurek

IT Systems Department, Jagiellonian University,  
Straszewskiego 27, 31-110 Cracow, Poland  
jjurek@wzks.uj.edu.pl

**Summary.** In the paper we present the method of the generalisation of a language sample for grammatical inference of quasi context-sensitive GDPLL( $k$ ) grammars. GDPLL( $k$ ) grammars and parsers have been developed as an efficient tool for syntactic pattern recognition: the grammars are characterised by very good discriminative properties and the parser for the grammars is of the linear computational complexity. Nevertheless, one of the main problems of practical application of GDPLL( $k$ ) grammars in syntactic pattern recognition systems consists in difficulties in defining the grammar from the sample of a pattern language. The method which we describe in the paper is an important element of the solution of this problem.

## 1 Introduction

One of the main problems in the syntactic pattern recognition area is the grammatical inference, i.e. automatical definition of a grammar from the sample of a pattern language. Although there are many grammatical inference methods already developed [1, 4, 5, 10, 11], there is still lack of general models of inferencing grammars of big generative/discriminative power (bigger than context-free grammars).

Quasi context-sensitive GDPLL( $k$ ) grammars are characterised by very good discriminative properties since their generative power is "almost" as big as context-sensitive grammars [2, 8]. We have introduced GDPLL( $k$ ) grammars and parsers as efficient tools for syntactic pattern recognition-based systems. The research into GDPLL( $k$ ) grammars and parsers have been started for the purpose of an on-line analysis of complex patterns representing trend functions related to the behavior of a very complex industrial-like equipment [2]. These trend functions have been treated in an analogous way, as it is made while applying syntactic pattern recognition for ECG or EEG analysis. GDPLL( $k$ ) grammars have been proven to be a suitable tool not only for describing the trend functions (because of their very good discriminative properties), but for analysing them as well (the GDPLL( $k$ ) parser is of linear computational complexity [6]). During a few last years we have been verifying the practical usefulness of GDPLL( $k$ ) grammars in several different application areas (eg. [3, 9]). Although we have achieved good results, we have noticed that the main problem of practical applications of GDPLL( $k$ ) grammars consists in difficulties in defining the grammar

from the sample of a pattern language. This is exactly the problem of grammatical inference, which we indicated at the beginning of this section as one of the main problems of syntactic pattern recognition.

In the paper we present the recent results of the research into grammatical inferencing of  $GDPLL(k)$  grammars: the method of generalisation of the sample of a language which is the first phase of the grammatical inference process. Section 2 contains basic definitions needed to discuss the method. The description of the model of the grammatical inference is included in section 3. Section 4 contains the specifications of algorithms. Conclusions are included in the final section.

## 2 Basic Definitions

Let us introduce a few basic definitions [2, 7, 8].

**Definition 1.** A *generalised dynamically programmed context-free grammar* is a six-tuple:  $G = (V, \Sigma, O, P, S, M)$ , where  $V$  is a finite, nonempty alphabet;  $\Sigma \subset V$  is a finite, nonempty set of terminal symbols (let  $N = V \setminus \Sigma$ );  $O$  is a set of basic operations on the values stored in the memory (assignment, addition, subtraction, multiplication);  $S \in N$  is the starting symbol;  $M$  is the memory (the memory is defined as such a memory structure that integer numbers can be stored in it and each element of the memory can be accessed in a constant time);  $P$  is a finite set of productions of the form:  $p_i = (\mu_i, L_i, R_i, A_i)$  in which  $\mu_i : M \longrightarrow \{TRUE, FALSE\}$  is the predicate of applicability of the production  $p_i$  defined with the use of operations ( $\in O$ ) performed over  $M$ ;  $L_i \in N$  and  $R_i \in V^*$  are left- and right-hand sides of  $p_i$  respectively;  $A_i$  is the sequence of operations ( $\in O$ ) over  $M$ , which should be performed if the production is to be applied. □

A derivation for generalised dynamically programmed grammars is defined in the following way. Apart from testing whether  $L_i$  occurs in a sentential form derived, we check the predicate of applicability of a production  $p_i$ . The predicate is defined as an expression based on variables stored in the memory. The evaluation of the predicate can need some calculations over integer values. If the predicate is true, we replace  $L_i$  with  $R_i$  and then we perform the sequence of operations over the memory. The execution of the operations changes the contents of the memory (memory variables). It is done with the help of arithmetical and assignment instructions. □

**Definition 2.** Let  $G = (V, \Sigma, O, P, S, M)$  be a generalised dynamically programmed context-free grammar. The grammar  $G$  is called a *Generalised Dynamically Programmed  $LL(k)$  grammar*,  $GDPLL(k)$  grammar, if: 1) the  $LL(k)$  condition of deterministic derivation is fulfilled, and: 2) the number of steps during derivation of any terminal symbol is limited by a constant. (Formal specifications of the two conditions is included in [2]). □

The main definitions needed to discuss the algorithm of generalisation of a language sample for grammatical inference GDPLL( $k$ ) grammars are the following:

**Definition 3.** Let  $A$  is a set of all (terminal) symbols which appear in the sample,  $N$  set of integer variables. *Polynomial specification of a language* is of the form:  $L_p(A, N) = S_i^{p_j(n_k)}$  where:  $p_j$  is a polynomial of a variable  $n_k \in N$ ; variable  $n_k$  can be assigned only values greater or equal 1 ( $n_k \geq 1$ );  $S_i$ , called *polynomial structure*, is defined in a recursive way:

- 1)  $S_i = (a_{i_1} \dots a_{i_r})$ , where  $a_{i_j} \in A$   
( $S_i$  is called a *basic* polynomial structure)  
or
- 2)  $S_i = (S_{i_1}^{p_{i_1}(n_{i_1})} \dots S_{i_r}^{p_{i_r}(n_{i_r})})$ , where  $S_{i_k}$  is defined as in 1) or 2).  
( $S_i$  is called a *complex* polynomial structure.) □

**Definition 4.** Let  $fs(L_p(A, N))$  be *the first symbol* of polynomial specification of a language, since for each polynomial specification of a language there is only one terminal symbol, which is the starting symbol of all words belonging to the language. Correspondingly, let  $ls(L_p(A, N))$  be *the last symbol* of polynomial specification of a language. □

**Definition 5.** Let us define *extended* polynomial specification of a language  $L_{ep}(A, N)$  in the following way:

- 1)  $L_{ep}(A, N) = L_p(A, N)$   
or
- 2)  $L_{ep}(A, N) = L_{ep}^1(A_1, N_1) L_{ep}^2(A_2, N_2) \dots L_{ep}^z(A_z, N_z)$  if  $N_i \cap N_j = \emptyset$  for each  $i \neq j$  and  $LS(L_{ep}^i(A_i, N_i)) \cap FS(L_{ep}^{i+1}(A_{i+1}, N_{i+1})) = \emptyset$  for each  $1 < i < z$ .  
Then  $A = \bigcup A_i$  and  $N = \bigcup N_i$  for  $i = 1, \dots, z$ ,  
or
- 3)  $L_{ep}(A, N) = L_{ep}^1(A_1, N_1) + L_{ep}^2(A_2, N_2) + \dots + L_{ep}^z(A_z, N_z)$  if  $N_i \cap N_j = \emptyset$  for each  $i \neq j$  and  $FS(L_{ep}^i(A_i, N_i)) \cap FS(L_{ep}^j(A_j, N_j)) = \emptyset$  for each  $i \neq j$ .  
Then  $A = \bigcup A_i$  and  $N = \bigcup N_i$  for  $i = 1, \dots, z$ ,

where  $FS(L_{ep}(A, N))$  — *the set of first symbols* of the extended polynomial specification of a language and  $LS(L_{ep}(A, N))$  — *the set of last symbols* of the extended polynomial specification of a language, are defined as follows:

- If  $L_{ep}(A, N) = L_p(A, N)$  then  $FS(L_{ep}(A, N)) = \{fs(L_p(A, N))\}$  and  $LS(L_{ep}(A, N)) = \{ls(L_p(A, N))\}$ .
- If  $L_{ep}(A, N) = L_{ep}^1(A_1, N_1) L_{ep}^2(A_2, N_2) \dots L_{ep}^z(A_z, N_z)$  then  $FS(L_{ep}(A, N)) = FS(L_{ep}^1(A_1, N_1))$  and  $LS(L_{ep}(A, N)) = LS(L_{ep}^z(A_z, N_z))$ .
- If  $L_{ep}(A, N) = L_{ep}^1(A_1, N_1) + L_{ep}^2(A_2, N_2) + \dots + L_{ep}^z(A_z, N_z)$  then  $FS(L_{ep}(A, N)) = FS(L_{ep}^1(A_1, N_1)) \cup \dots \cup FS(L_{ep}^z(A_z, N_z))$  and  $LS(L_{ep}(A, N)) = LS(L_{ep}^1(A_1, N_1)) \cup \dots \cup LS(L_{ep}^z(A_z, N_z))$ . □

### 3 The Idea of the Grammatical Inference

Our approach to the inferencing GDPLL( $k$ ) grammars is based on the following method [7]. We divide the inference process into two phases. The first phase is responsible for extraction of the features of the sample, and generalisation of the sample. The extended polynomial specification of a language is obtained as the result of this phase (see: Definition 5). In the second phase, a GDPLL( $k$ ) grammar is generated on the basis of the extended polynomial specification of the language. Both phases are independent of each other. The second phase is well-defined, i.e. the result of this phase is strictly determined by the input data (an extended polynomial specification of language). On the contrary, the first phase is not well-defined since there may be many approaches to the generalisation of the sample and the "quality" of an approach depends on a particular application and a particular sample of the language. It makes the problem of the construction of suitable algorithms for this phase especially difficult.

Our preliminary result of the research into grammatical inference of GDPLL( $k$ ) grammars has been described in [7]. In that paper we have presented the algorithm of automatic generation of a GDPLL( $k$ ) grammar from polynomial specification of a language (as the main algorithm of the second phase).

Now, we are going to present the method of creating the extended polynomial specification of a language form the sample (i.e. the first phase of grammatical inference: the extraction of the features of the sample, and the generalisation of the sample). Let us notice that our method is canonical. It can be used for the detection of basic features only, and for relatively simple generalisation of the sample. The canonical character of the method may broaden the area of its possible applications.

### 4 Generalisation of the Sample of a Language

Let the sample of a language be a set of  $m$  words:  $Sample = \{w_1, \dots, w_m\}$ . Let  $\Sigma$  be a common alphabet for all words in the sample. Words belonging to the sample will be written in the form:  $w = a_1^{n_1} a_2^{n_2} \dots a_k^{n_k}$ , assuming that  $a_i \neq a_{i+1}$  for  $1 \leq i < k$ .

**Definition 6.** Let us consider two words:  $w_1 = a_1^{p_1} a_2^{p_2} \dots a_k^{p_k}$  and  $w_2 = b_1^{q_1} b_2^{q_2} \dots b_l^{q_l}$ . We say that words  $w_1$  i  $w_2$  are *sequentially equivalent*, if the following conditions are fulfilled:  $k = l$  and  $a_i = b_i$  for  $i = 1, \dots, k$ . □

**Definition 7.** *The template* of an extended polynomial specification is an expression constructed accordingly to the definition of extended polynomial specification  $L_{ep}$ , which is built with sets of sequentially equivalent words instead of polynomial specifications  $L_p$ . □

The process of construction of the extended polynomial specification of a language can be described by two algorithms. The first one is responsible for the

construction of the template of the extended polynomial specification. The second one is responsible for the generalisation of each set of sequentially equivalent words (in the template) in the form of a polynomial specification.

**Algorithm 1.** Let  $Sample = \{w_1, \dots, w_m\}$ . The algorithm of the construction of the template of extended polynomial specification of the language from  $Sample$  is the following. Let  $V\{v_1, \dots, v_n\}$  be a set which is a variable in the template of an extended polynomial specification which is being constructed, where  $v_1, \dots, v_n$  are words or fragments of words belonging to  $Sample$ .

*Step 1.* Initially, let the template of an extended polynomial specification be the variable:  $V\{w_1, \dots, w_m\}$ .

*Step 2.* While there is such a set  $V$  in the template, which is not the set of sequentially equivalent words, we transform the template according to two rules:

- a)  $V\{w_{s_1} a_1 w_{r_1}, \dots, w_{s_n} a_n w_{r_n}\} := V\{w_{s_1}, \dots, w_{s_n}\} V\{a_1 w_{r_1}, \dots, a_n w_{r_n}\}$   
 if  $w_{s_i}$  are sequentially equivalent (for  $i = 1, \dots, n$ ) and there are such symbols  $a_i$  and  $a_j$  different than the last symbol of  $w_{s_1}$  that  $a_i \neq a_j$ .
- b)  $V\{a_1 w_1, \dots, a_n w_n\} := V\{a_{1_k} w_{1_k}, \dots, a_{1_k} w_{1_k}\} + V\{a_{2_k} w_{2_k}, \dots, a_{2_k} w_{2_k}\} + \dots + V\{a_{s_k} w_{s_k}, \dots, a_{s_k} w_{s_k}\}$   
 if for any two symbols  $a_{x_i}, a_{x_j} : a_{x_i} = a_{x_j}$ , and for any two symbols  $a_{x_i}, a_{y_j}$ , where  $x \neq y : a_i \neq a_j$

After all needed iterations of step 2 (at most as much times as it the length of the longest words in sample), the template is constructed in which all  $V$  sets are sets of sequentially equivalent words. □

**Algorithm 2.** Let  $P = \{w_1, \dots, w_q\}$  is a set of sequentially equivalent words. Let the word  $w_i \in P$  is of the form:  $w_i = a_1^{n_{i1}} a_2^{n_{i2}} \dots a_k^{n_{ik}}$ . The canonical algorithm of the construction of polynomial specification  $L_p(A, N)$  for  $P$  consists of three steps:

*Step 1.* For  $L_p(A, N)$  we define:  $A = \{a_1, \dots, a_k\}$ , and  $N = \{x_1, \dots, x_k\}$ . We assume that  $L_p(A, N) = a_1^{p_1(x_1)} \dots a_k^{p_k(x_k)}$ .

*Step 2.* For each  $j = 1, \dots, k$  we search for the linear function of variable  $x_j$ , which describes dependencies between numbers  $n_{i_j}$  (where:  $i = 1, \dots, q$ ). The method is trivial so we do not describe it here. Let the function be of the form:  $p_j(x_j) = d_j x_j + r_j$ .

*Step 3.* We look for dependencies between  $x_j = x_1, \dots, x_k$  allowing to reduce  $N$  set. For each pair  $x_g$  and  $x_h$  (where  $g = 1, \dots, k, h = 1, \dots, k$  and  $h \neq g$ ) we perform:

1. Let us evaluate variables  $x_g$  and  $x_h$  for word  $w_1$  accordingly to the equations:  

$$x_g = \frac{n_{1g} - r_g}{d_g} \text{ and } x_h = \frac{n_{1h} - r_h}{d_h}. \text{ Let } x_g \leq x_h. \text{ Let } s = x_h - x_g.$$

2. We test if  $x_h = x_g + s$  for each other words in  $P$ , i.e. if  $n_{i_h} = d_h(x_g + s) + r_h$  for  $i = 2, \dots, q$ .
3. If the result of the test is negative — we do not find any dependencies between  $x_h$  and  $x_g$ . Otherwise — we store function  $p_h(x_g) = d_h x_g + (d_h s + r_h)$  for the index  $h$ , and we delete variable  $x_h$  from  $N$  set.

As the outcome of the algorithm we get polynomial specification  $L_p(A, N)$  such that all words from  $P$  belong to  $L_p(A, N)$ , as well as all other words which are considered to be "similar".  $\square$

The extended polynomial specification of the language is the result of the application of Algorithm 1 and Algorithm 2. We get the specification by inserting polynomial specifications (generated by Algorithm 2) to the template of the extended polynomial specification (generated by Algorithm 1). The extended polynomial specification is the input for the second phase of the grammatical inference.

## 5 Concluding Remarks

In the paper we have presented the recent results of our research into grammatical inference of GDPLL( $k$ ) grammars. We have developed a method of the generalisation of the sample of a language in the form of an extended polynomial specification. An extended polynomial specification can be then a base for automatic generation of a GDPLL( $k$ ) grammar [7].

The results are important and promising if we consider the applications of the model. GDPLL( $k$ ) grammars are characterised by very good discriminative properties (their generative power is "almost" as big as context-sensitive grammars) and they are computationally efficient (the GDPLL( $k$ ) parser is of linear computational complexity). These two features make GDPLL( $k$ ) grammars a very good tool for building syntactic pattern recognition systems (this fact has been confirmed by our practical experiences during a few last years). The grammatical inference method for GDPLL( $k$ ) grammars is a new tool which may significantly broaden the scope of the applications.

Let us notice, that although the results are promising, there are still many open research issues. First of all we must improve the algorithm of the automatic generation of GDPLL( $k$ ) (described in [7]) and make it able to handle not only polynomial specifications but *extended* polynomial specifications as well. The work on the solution for this problem is very advanced. Secondly, we are going to include *a negative sample* (i.e. a sample of words which do not belong to the language) into the process of grammar inference. Last but not least, our method needs a thorough practical evaluation. We are going to test it in the environment of several different applications ([3, 9]) where the need of a grammatical inference algorithm is particularly noticeable.

## References

1. Alquezar R, Sanfeliu A (1997) Recognition and learning of a class of context-sensitive languages described by augmented regular expressions. *Pattern Recognition* vol. 30 no. 1: 163–182
2. Flasiński M, Jurek J (1999) Dynamically Programmed Automata for Quasi Context Sensitive Languages as a Tool for Inference Support in Pattern Recognition-Based Real-Time Control Expert Systems. *Pattern Recognition* vol. 32 no. 4: 671–690
3. Flasiński M, Reroń E, Jurek J, Wójtowicz P, Atlasiewicz A (2005) On the construction of the syntactic pattern recognition-based expert system for auditory brainstem response analysis. In: *Computer Recognition Systems*. Springer, Berlin Heidelberg New York
4. Higuera De La C (2000) Current Trends in Grammatical Inference. *Lecture Notes in Computer Science* vol. 1876: 28–31
5. Higuera De La C (2005) A bibliographical study of grammatical inference. *Pattern Recognition* vol. 38 no. 9: 1332–1348
6. Jurek J (2000) On the Linear Computational Complexity of the Parser for Quasi Context Sensitive Languages. *Pattern Recognition Letters* no. 21: 179–187
7. Jurek J (2004) Towards grammatical inferencing of GDPLL( $k$ ) grammars for applications in syntactic pattern recognition-based expert systems. *Lecture Notes in Computer Science* no. 3070: 604–609
8. Jurek J (2005) Recent developments of the syntactic pattern recognition model based on quasi-context sensitive languages. *Pattern Recognition Letters* no. 26: 1011–1018
9. Jurek J, Peszek T (2005) On the use of syntactic pattern recognition methods, neural networks, and fuzzy systems for short-term electrical load forecasting. In: *Computer Recognition Systems*. Springer, Berlin Heidelberg New York
10. Sakakibara Y (1997) Recent Advances of Grammatical Inference. *Theoretical Computer Science* vol. 185 no. 1: 15–45
11. Sakakibara Y (2005) Learning contex-free grammars using tabular representations. *Pattern Recognition* vol. 38 no. 9: 1372–1383



---

# Pattern Classification Using Efficient Linear Classifiers with Small Number of Weights

Magdalena Topczewska

Białystok Technical University, Faculty of Computer Science  
magda@ii.pb.bialystok.pl

**Summary.** In this paper we compare two approaches to the estimation of parameters describing linear classifiers considering a feature selection possibility for multidimensional data. The parameters are determined by minimization of convex and piecewise linear penalty functionals. Optimization is performed using a base exchange strategies. We assess numerical efficiency, classification accuracy, and the dimension of the subspaces containing the found solutions.

## 1 Introduction

Many of the pattern recognition tasks refer to classification and regression problems. If objects are described as  $n$ -dimensional vectors in some input space and belong to two or more classes, they form *learning sets*. Classification problem is then tantamount to partitioning the input space into regions related to classes and a new object, for which the label of the class is unknown, can be classified on the basis of region it is located in.

There are several ways in which linear classifiers can be found. The most popular nowadays is the *Support Vector Machines* method (*SVM*) [6, 3]. Even if it is reasonable, it has some disadvantages. First, it might be not preferred in applications requiring great classification speed, because of the large number of support vectors [7, 4]. The second disadvantage of using this method can be the fact that there is no opportunity to influence on the number of nonzero parameters (weights), describing decision boundaries. Because of that we propose and test different formulations of linear classifiers that allow to a *feature selection*. The next advantage of these classifiers is that it can be formulated as a linear programming task and very fast algorithms to solve it can be used.

The remainder of the paper is organized as follows. In section 2, the brief problem statement is given. In section 3, well-known and new linear classification rules based on the convex and piecewise linear penalty functions are described. In section 4, tests on real-world databases are presented. In section 5, concluding remarks are described.

## 2 Problem Statement

Let us assume that a set containing  $m$  objects is given. Each object consists of a pair  $(\mathbf{x}_i, y_i)$ , where  $\mathbf{x}_i$  ( $i = 1, \dots, m$ ) is a so-called *feature vector* or a point

in  $n$ -dimensional *feature space*. The components of the feature vector can have real or binary values ( $\mathbf{x}_i \in R^n$ ,  $\mathbf{x}_i \in \{0, 1\}^n$ ). An  $y_i$  value indicates the label of a class to which the  $i$ -th object is assigned. In the two-classes case the bipolar representation is preferred ( $y_i \in \{-1, +1\}$ ).

A location of the optimal hyperplane  $H(\mathbf{w}^*, b^*)$ , separating objects from two classes,

$$\begin{aligned}
 h(\mathbf{w}, b; \mathbf{x}) &= \mathbf{w}^T \mathbf{x} + b: R^n \rightarrow R \\
 H &= \{\mathbf{x} \in R^n: h(\mathbf{x}) = 0\}
 \end{aligned}
 \tag{1}$$

can be found by solving optimization problem and fixing the optimal values for vector  $\mathbf{w}$  – normal to the hyperplane and the threshold  $b$ .  $\frac{b}{\|\mathbf{w}\|}$  is the perpendicular distance from the hyperplane to the origin and  $\|\mathbf{w}\|$  is the Euclidean norm of  $\mathbf{w}$ . All points which satisfy  $\mathbf{w}^T \mathbf{x} + b = 0$  lie on the hyperplane.

Indicating the class of objects with label  $+1$  by  $C^+$  and the class of objects with label  $-1$  by  $C^-$ , the definition can be quoted [2].

**Definition 1.** *The learning sets are linearly separable in the  $n$ -dimensional feature space if and only if there exists the hyperplane  $H(\mathbf{w}, b)$  which completely separates these sets*

$$\exists \mathbf{w}, b \quad (\forall \mathbf{x}_i \in C^+) \quad \mathbf{w}^T \mathbf{x}_i + b > 0 \quad \wedge \quad (\forall \mathbf{x}_i \in C^-) \quad \mathbf{w}^T \mathbf{x}_i + b < 0 \tag{2}$$

According to the definition 1 all vectors belonging to the class  $C^+$  are located on the positive side of the hyperplane  $H(\mathbf{w}, b)$  and satisfy  $\mathbf{w}^T \mathbf{x} + b > 0$ , similarly, all vectors belonging to the class  $C^-$  are located on the negative side of the hyperplane and satisfy  $\mathbf{w}^T \mathbf{x} + b < 0$ .

### 3 Criterion and Methods Description

Evaluation of the resulting classifier is based on the calculation of the number of objects, which are located on the incorrect side of the hyperplane. The objective function can be expressed as:

$$\Phi(\mathbf{w}, b) = \sum_{i=1}^m |sgn(\mathbf{w}^T \mathbf{x}_i + b) - y_i| \tag{3}$$

and the minimum of (3) should be found.

In view of the fact, that the formula (3) is discontinuous, it is a difficult optimization problem. Thus other approaches, regularizing (3), might be used.

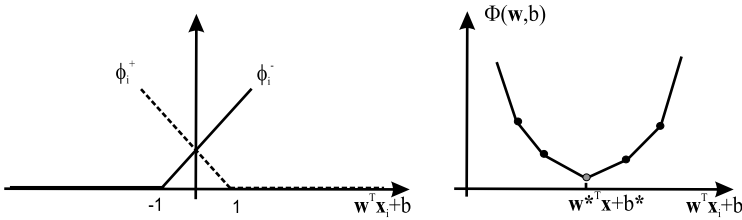
*Convex and piecewise linear functions*

The convex and piecewise linear objective function can be represented as a sum of misclassification error values of all objects

$$\Phi(\mathbf{w}, b) = \sum_{i=1}^m \tau(\delta_i - y_i(\mathbf{w}^T \mathbf{x}_i + b)) \tag{4}$$

where  $\tau(\cdot)$  is a positive part of a number and  $\delta_i$  is a margin. The formulation (4) will be called *PLPF* (*piecewise linear penalty functions*).

For the object located on the correct side of the hyperplane misclassification error value equals zero. For the object situated on the incorrect side of the hyperplane the misclassification error value increases linearly with the distance between the object and the hyperplane. The  $i$ -th component of the function (4) for a single object is then piecewise linear and convex and the sum of such functions constitutes the piecewise linear and convex function *PLPF* (4) (Fig. 1). Setting into the formula (4) margins  $\delta_i$  leads to punishing objects, situated on the correct side of the hyperplane, for which distance to the decision boundary is smaller than the value  $\delta_i$ . For the  $i$ -th object the margin  $\delta_i = 1$  is assumed.



**Fig. 1.** Convex and piecewise linear functions:  $\phi_i^+$  for the object from the class  $C^+$ ,  $\phi_i^-$  for the object from the class  $C^-$  and the objective function  $\Phi(\mathbf{w}, b)$

Let us take into account the second objective function created by adding to the formula *PLPF* (4) the sum of the weights

$$\Phi(\mathbf{w}, b) = \sum_{i=1}^m \tau(1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)) + \lambda \sum_{i=1}^m |w_i| \tag{5}$$

where  $\lambda$  is a small number. The formulation (5) will be called *PLPF* <sub>$L_1$</sub> .

The function retains the property of being convex and piecewise linear. The additional beneficial aspect can be the fact that minimization of the formula *PLPF* <sub>$L_1$</sub>  (5) with the parameter  $\lambda$  tuned might be used to decrease in a number of nonzero weights. This is tantamount to a feature selection.

*Methods*

There are two algorithms for solving the problems *PLPF* (4) and *PLPF* <sub>$L_1$</sub>  (5). The first one, called *Sekwem*, finds the parameters of the separating hyperplanes by minimizing *PLPF* (4) or *PLPF* <sub>$L_1$</sub>  (5) using the algorithm similar to an exchange base technique [2, 1]. The algorithm is iterative and in each iteration the direction of the steepest decrease of the objective function is chosen. Then the algorithm moves along this direction until there is no more objective function decrease. In the next step the new direction is chosen and the process is repeated. If there is no decrease of the function in any direction, the algorithm stops and the minimum of the function is found.

The second algorithm, called *Genet* is based on the *Sekwem* algorithm. The main difference is that the enlargement of the feature space is progressive. In the first step of the algorithm one or several variables for calculations might be chosen. In every next step only one variable, which provides the largest decrease of the objective function, is added. If there is no such variable, the process is stopped and the optimal hyperplane, described by the set of nonzero weights, is found.

In both algorithms the identity matrix  $I^{n \times n}$  is taken as the initial basis and inserted into the data. If we have a *long data* case, when  $n > m$ , it might give rise to considerable enlargement of the data. Current implementation of these algorithms does not allow to perform operations on sparse matrices. Thus the random access memory for all data and the initial matrix must be allocated. For multidimensional data the calculation time can draw out.

*New formulations*

Computing the error values in *PLPF* (4) or *PLPF*<sub>L<sub>1</sub></sub> (5), constraints (6) are used:

$$\delta_i - y_i(\mathbf{w}^T \mathbf{x}_i + b) \leq 0. \tag{6}$$

They might be expressed as the following matrix:

$$\mathbf{A}\mathbf{w} - \mathbf{y}b - \mathbf{I}\mathbf{u} = \mathbf{1} \tag{7}$$

where  $\mathbf{A}$  is a matrix of feature vectors,  $\mathbf{I} \in R^{n \times m}$  is an identity matrix,  $\mathbf{1} \in R^m$  is an identity vector and  $\mathbf{u} \in R^m$  is a vector of slack variables.

The non-negative variable  $\mathbf{u}$  can be split into the two parts – positive and negative and the appropriate part is added to the objective function. This is a new formulation of the problem, which forms a linear optimization task and is equivalent to *PLPF* (4):

$$\begin{aligned} \mathbf{A}\mathbf{w} - \mathbf{y}b - \mathbf{I}\mathbf{u}^+ + \mathbf{I}\mathbf{u}^- &= \mathbf{1} \\ \Phi = \mathbf{c}^T \mathbf{u}^- &= \min! \\ \mathbf{u}^+, \mathbf{u}^- &\geq 0 \end{aligned} \tag{8}$$

The formulation (8) (called *SFPF*) has almost the standard form of linear programming task, but the variables  $\mathbf{w}$  and  $b$  have unrestricted signs. They can be split into two parts and through this operation non-negative values for them are required. To the objective function an additional part is added. The new formulation, equivalent to *PLPF*<sub>L<sub>1</sub></sub> (5) is expressed as

$$\begin{aligned} \mathbf{A}\mathbf{w}^+ - \mathbf{A}\mathbf{w}^- + \mathbf{y}b^+ - \mathbf{y}b^- - \mathbf{I}\mathbf{u}^+ + \mathbf{I}\mathbf{u}^- &= \mathbf{1} \\ \Phi = \mathbf{E}^T \mathbf{w}^+ + \mathbf{E}^T \mathbf{w}^- + \mathbf{c}^T \mathbf{u}^- &= \min! \\ \mathbf{w}^+, \mathbf{w}^-, b^+, b^-, \mathbf{u}^+, \mathbf{u}^- &\geq 0 \end{aligned} \tag{9}$$

The formulation (9) will be called *SFPF*<sub>L<sub>1</sub></sub>.

Describing the problem as *SFPF* (8) or *SFPF*<sub>L<sub>1</sub></sub> (9) does not lead to the increase in the amount of data. However, the dimension of data is enlarged by two identity matrices. For the long data case this formulation might be much more efficient.

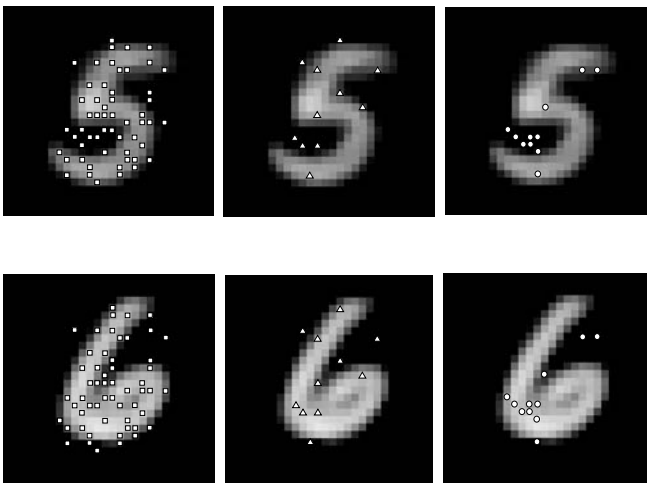
Because the problem is linear, classical methods for linear programming tasks can be used, i.e. the *simplex method*. Nevertheless, when we have multidimensional data, moving along the edges of the polytope until the vertex of the optimum solution is reached, requires a large amount of changes of adjacent vertices. A better solution is to apply solvers for large-scale problems.

### 4 Numerical Results

The methods were tested on both artificial and real-world datasets. Due to a space limitation, the experimental results only on few real-world datasets are presented. The experiments were performed on computer with Intel Core Duo T2500 2.GHz processor and 1GB RAM.

**Table 1.** Results for  $PLPF$ ,  $PLPF_{L_1}$ ,  $SFPF$  and  $SFPF_{L_1}$  methods for handdigits data

dataset		Metody			
		PLPF	$PLPF_{L_1}$	SFPF	$SFPF_{L_1}$
handdigits_56	$n_{nw}$	52	11	514	11
	$\Phi$	0	$1.49758e - 006$	$9.5355e - 005$	$7.4774e - 005$
	$Q_{zb\_ucz}$	100%	100%	100%	100%
	$Q_{zb\_test}$	92.25%	93.25%	94%	93.5%
handdigits_08	$n_{nw}$	33	14	480	7
	$\Phi$	0	$9.5442e - 005$	0.00021263	0.0001435
	$Q_{zb\_ucz}$	100%	100%	100%	100%
	$Q_{zb\_test}$	93.25%	98%	98.25%	98.25%
handdigits_17	$n_{nw}$	15	5	472	9
	$\Phi$	0	$4.84285e - 007$	$9.5968e - 005$	0.00019064
	$Q_{zb\_ucz}$	100%	100%	100%	100%
	$Q_{zb\_test}$	92.25%	95%	98%	97%

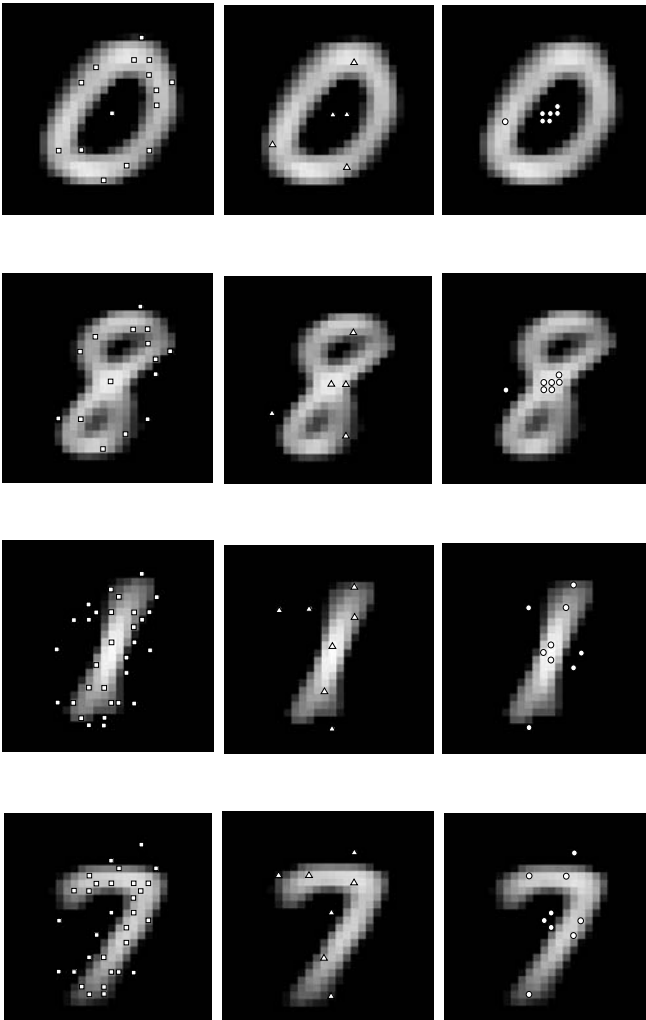


**Fig. 2.** Average images of handwritten digits with important variables marked, squares -  $PLPF$  method, triangles -  $PLPF_{L_1}$  method, circles -  $SFPF_{L_1}$  method (a)

*Experiment*

In the experiment data sets containing collection of handwritten digits were chosen. Every digit was described as a matrix of  $28 \times 28$  pixels in grayscale palette. Three pairs of digits were classified – 0 and 8, 1 and 7, 5 and 6. In this article the results are compared with the results published in [5].

The results are presented in the Table 1. As we can see, the new formulations  $SFPF$  (8) and  $SFPF_{L_1}$  (9) give better results in classification accuracy than (4) and (5). However, the number of nonzero parameters describing the separating hyperplane is comparable only between  $PLPF_{L_1}$  (5) and  $SFPF_{L_1}$  (9) methods.



**Fig. 3.** Average images of handwritten digits with important variables marked, squares -  $PLPF$  method, triangles -  $PLPF_{L_1}$  method, circles -  $SFPF_{L_1}$  method (b)

The largest number of nonzero weights is observed for *SFPF* (8) method in all cases. Because of these, the variables important for classification decision for *SFPF* (8) are not presented on the figure (2). The *PLPF* (4) - the criterion function equivalent to *SFPF* (8) is better for feature selection purpose but the classification accuracy is a lower value than for *SFPF* (8). The nonzero variables selected in both methods are only partially covered. It seems that the choice of variables taking part in classification decision is more intuitive for *SFPF* (8).

It should also be mentioned that times of calculations for *SFPF* (8) and *SFPF*<sub>L<sub>1</sub></sub> (9) were several or several dozen times shorter than for *PLPF* (4) and *PLPF*<sub>L<sub>1</sub></sub> (5).

## 5 Conclusions

In this paper approaches to the estimation of the linear classification bounds were presented. A feature selection possibility for multidimensional data is considered. The parameters are determined by minimization of convex and piecewise linear penalty functionals. The new formulations as linear programming tasks enable to use very efficient algorithms for finding best separating hyperplanes.

For multidimensional data, for which the number of attributes is much bigger than the number of observations, the time of finding the solution is much shorter for *SFPF* and *SFPF*<sub>L<sub>1</sub></sub> than for *PLPF* and *PLPF*<sub>L<sub>1</sub></sub>. The application of solvers for large-scale tasks and techniques for sparse matrices can reduce the time of calculations considerably.

The smallest number of nonzero parameters describing separating hyperplane can be achieved by using *PLPF*<sub>L<sub>1</sub></sub> or *SFPF*<sub>L<sub>1</sub></sub> methods, but the *SFPF*<sub>L<sub>1</sub></sub> method gives separating hyperplanes with better accuracy and with comparable number of nonzero weights.

## Acknowledgment

This work was supported by the Białystok Technical University grant no W/WI/1/05.

## References

1. Bobrowski L (1991) Design of piecewise linear classifiers from formal neurons by some basis exchange technique, *Pattern Recognition*, 24(9), pp. 863-870, 1991.
2. Bobrowski L (2005) Exploratory data analysis based on convex and piecewise linear (CPL) criterion functions (in Polish), Technical University Białystok, 2005.
3. Burges CJC (1998) A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery* 2:121-167, 1998.
4. Chang CC, Lin CJ (2001) Training nu-support vector classifiers: Theory and algorithms, *Neural Computation*, 13: 2119-2147, 2001.

5. Topczewska M, Lukaszuk T, Bobrowski L (2006) Modelling of long data sets with examples of applications (in Polish), XIII Workshop of Polish Society of Computer Simulation, 2006.
6. Vapnik VN (1998) Statistical Learning Theory. J. Wiley, New York, 1998.
7. Yu H and Yang J and Han J (2003), Classifying large data sets using SVMs with hierarchical clusters. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.



---

# Markov Chain Model for Tree-Based Genetic Algorithm Used in Computer Aided Design

Anna Paszyńska<sup>1</sup>, Marian Jabłoński<sup>2</sup>, and Ewa Grabska<sup>3</sup>

<sup>1</sup> International School of Business, Design and Technology, Bielsko-Biala, Poland  
apaszynska@hotmail.com

<sup>2</sup> Jagiellonian University, Institute of Computer Science, Krakow, Poland

<sup>3</sup> Jagiellonian University, The Faculty of Physics, Astronomy and Applied Computer Science, Krakow, Poland

**Summary.** In this paper, we present the hierarchical chromosome based genetic algorithm, which is used in design systems to solve complex design problems. The main goal of our work is creating of the markov chain model for the algorithm. The model contains a method of coding of the artifacts, represented by different length trees, by means of fixed length binary strings and a method of the hierarchical crossover modeling. Presented formalization is used for finding the transition matrix and the investigation of ergodicity and asymptotic properties of genetic algorithms. It also follows from the model that each population can be found with the probability equal to one within the finite number of steps.

## 1 Introduction

Evolutionary algorithms are often used as the search algorithms for the best solutions in the design space. The fixed length representation and the traditional genetic operators used in simple genetic algorithm (SGA), are insufficient in many design problems. Many evolutionary algorithms require more complicated representation and genetic operators. Some examples of such algorithms are: tree based genetic programming, graph-based genetic algorithms [2] and hierarchical chromosome based genetic algorithms [1, 4]. The main advantage of design systems with hierarchical chromosome based evolutionary algorithms is the ability of solving of a design problem even if the optimal number of components is not known. An application of the such a design system to solving of exemplary design problem - optimization of platform shape - was described in [4]. The problem consists in finding the optimal shape of the platform with minimal overall volume and the stress value not greater then the maximum admissible value, when the platform is under the external load. The optimal shape of the platform is found by changing number, location and shape of the components of the platform.

In spite of the numerous applications, still little is known on mathematical models and theoretical results of evolutionary algorithms. There are many works based on Schema Theory. Nevertheless, the Schema Theory for the genetic search description can model only some aspects of evolutionary algorithms. The other approach to the formalization of GA is based on the microscopic Markov chain

models, such as VoseŠs model of SGA [6, 3]. In this paper, we present a Markov chain model for tree based GA. The model contains a method of coding of a tree by means of binary string and a method of modeling the hierarchical crossover. Presented formalization is used for finding the transition matrix and the investigation of ergodicity and asymptotic properties of GA. It also follow from the model that each population can be found with the probability equal to one within the finite number of steps. Presented formalization creates base for research on different types of genetic algorithms based on hierarchical structures. This theoretical approach provides an excellent framework for studying the dynamics of genetic algorithms.

## 2 Hierarchical Chromosome Based GA

The representation of objects (phenotypes) is one of the most important elements of the design system. The assumed representation defines the potential solution space generated by the system. It has an influence on the genotypeŠs size and complexity. In our works we consider phenotypes which are three dimensional objects. Following BentleyŠs conclusions [1], the Spatial Partitioning is used to represent solids. The representation consists in partitioning each solid (potentially solution of a design problem) into a number of smaller components that do not intersect. In our design problem, the Clipped Stretched Cube representation is used. A primitive here is a cuboid intersected by a plane defined relatively to its center. Each primitive is described by nine parameters: three coordinates of the center of the primitive, its height, width and depth, and three parameters defining the intersection plane. Each phenotype has corresponding hierarchy tree. An example of a 3D object and the hierarchical chromosome coding it are shown in fig.1. The exemplary object consists of 9 components, each of them is described by nine genes, each gene is coded as the binary string. The components are grouped into the classes in spite of their design meaning. In the presented example there are three classes.

The hierarchy tree based genetic algorithm can be described according to the scheme  $(\mu, \lambda)$ , where  $\mu = \lambda$ . The reproduction in the algorithm is done

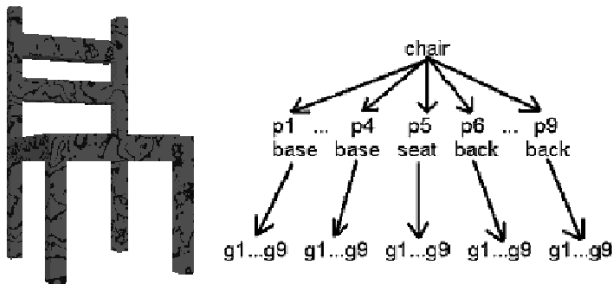


Fig. 1. An example of a 3D object and the hierarchical chromosome coding it

by fitness proportional selection and succession is done by elite selection, where  $\#Elite = \mu$ . The mutation and hierarchical crossover [1, 3] are used here.

### 3 Definition of the Artifact Space and the Genotype Space

The main goal of design is to define of the potential solutions space. The potential solutions are represented as models of graphical artifacts. We will call that space the artifact space. In many design applications the models of graphical artifacts are created from the transformed copies of preliminary shapes called primitives. Let  $F$  be a set of transformation from  $\mathbf{R}^n$  into  $\mathbf{R}^n$ , where each of them transforms bounded subsets into bounded subsets. Each element  $f \in F$  is called acceptable transformation.

**Definition 1.** Let  $F$  be a set of acceptable transformations from  $\mathbf{R}^n$  to  $\mathbf{R}^n$ . Let  $S$  be a set of bounded subsets in  $\mathbf{R}^n$ . If  $f(p) = q$  for  $f \in F, p, q \in S$  implies  $p = q$  than elements of  $S$  are called primitives under  $F$ . Artifact is a sum of transformed primitives called components. Each component is represented as pair  $(p, f)$  where  $p$  denotes primitive and  $f$  is the transformation.

**Definition 2.** Let  $S$  be the primitive set under the set  $F$  of acceptable transformations. Let  $Q$  be the artifact structure. Artifact  $A$  for  $Q$  is defined by the following formula:

$$A = \bigcup_{(p,f) \in Q} f(p) \tag{1}$$

**Definition 3.** Let  $S$  be the primitive set under the set  $F$  of acceptable transformations from  $\mathbf{R}^n$  into  $\mathbf{R}^n$ . Artifacts space  $R$  is defined as follows:

$$A = \{A : A = \bigcup_{(p,f) \in Q} f(p), \text{ where } Q \in P(S \times F)\} \tag{2}$$

**Definition 4.** Let  $R$  be the artifact space. Components class set for the artifact space  $R$  is defined by following formula:

$$K = \bigcup_{j=1}^d K_j \text{ where } \forall j \in \{1, \dots, d\} K_j \subset \{s : \exists A \in R : s \in A\} \tag{3}$$

Let  $P_A$  be the space of all artifacts such that:

- the number of components of each artifact  $B \in P_A$  is not higher than  $M_A$ ,
- classes  $K_1^B, K_2^B, \dots, K_d^B$  of components of artifact  $B \in P_A$  fulfill following conditions:  $\#K_1^B \leq N_1, \#K_2^B \leq N_2, \dots, \#K_d^B \leq N_d; N_1, N_2, \dots, N_d \in \mathcal{N}; \sum_{i=1}^d N_i \leq M_A$
- each component is described by  $p$  parameters (genes),
- each gene is coded by the binary string of the length  $q$ .

The space  $P_A$  is called parameterized artifacts space. Let  $B \in P_A$  be an artifact,  $n_i^B$  the number of components in class  $K_i$  for  $i \leq d_B$ , where  $d_B$  is the number of classes of the artifact B. Artifact B can be represented by the tree presented in fig.2. Here  $K_{i_1}, \dots, K_{i_{d_B}}$  denote classes  $s_j^i$  denotes j-th component of the i-th class,  $(gen_j^i)_k$  denotes k-th gene of the j-th component of the i-th class and  $b_k$  is the k-th bit.

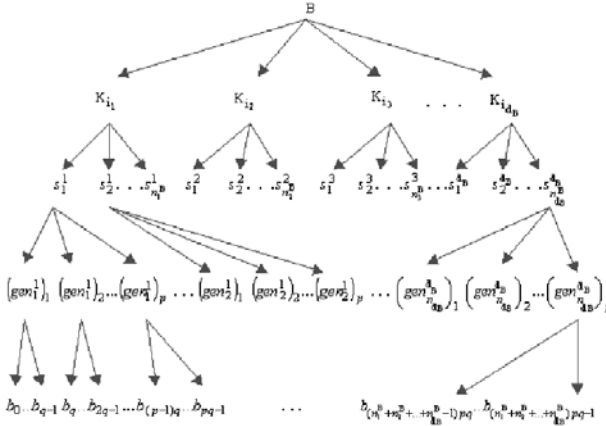


Fig. 2. An example of the hierarchy tree

**Definition 5.** The null component is the component with all genes coded as strings consisting only of zeros.

**Definition 6.** The universal hierarchy tree for artifact  $B \in P_A$  is a tree which was built from the hierarchy tree for the artifact for artifact B by adding  $N_i - n_B^i$  null components, for each class  $i$  such as  $N_i - n_B^i > 0$ . Trees with all components equal, represent the same artifact, independently from the location of the components.

**Definition 7.** The genotype space  $U_m$  for  $m = 2^l$  where  $l = M_A p q$ , is defined as  $U_m = (Z_2)^l$ .

**Definition 8.** The code of the artifact from the space  $P_A$  is the binary string  $x = (b_0, b_0, \dots, b_{l-1}) \in U_m$  from the lower level of the universal hierarchy tree for the artifact.

## 4 Markov Chain Model for Hierarchical Chromosome Based GA

The presented model was created to describe hierarchical chromosome based genetic algorithm. Hierarchical crossover will be modeled by the one-points crossover with the crossover point situated between the components.

Let us denote the set of all possible r-elements population by  $\Omega_r$ , the arbitrary probability space by  $\Omega$  and the number of the occurrences of the k-th element in the j-th population by  $r_k^j$ .

**Definition 9.** Let  $\eta : \Omega \rightarrow U_m$  be the random variable and  $mut : U_m \times U_m \rightarrow U_m$  be the function defined as follows:

$$mut(x, y) = x \oplus y \quad (4)$$

Mutation is the random operator defined as:  $mut(i, \eta)$ . We will consider only the probability distribution  $\eta$  defined as follows:

$$P(\eta = i) = \mu_i = a^{\mathbf{1}^T i} (1 - a)^{l - \mathbf{1}^T i} \quad (5)$$

where  $\mathbf{1}^T i$  is the number of the one's in the binary code of  $i$ ,  $l - \mathbf{1}^T i$  is the number of the zero's in the binary code of  $i$  and  $a \in (0, 1/2)$ , is the mutation rate.

**Definition 10.** Let  $\zeta : \Omega \rightarrow U_m$  and  $\kappa : \Omega \rightarrow \{0, 1\}$  be the random variables and let function  $K : U_m \times U_m \times U_m \times 0, 1 \rightarrow U_m$  be defined as follows:

$$K(x, y, z, u) = x \otimes z^u \oplus z^{1-u} \otimes y \quad (6)$$

where  $z^u = z$  if  $u = \mathbf{0}$  and  $z^u = \bar{z}$  if  $u = \mathbf{1}$ . Crossover is the random operator defined as:  $K(x, y, z, \kappa)$ . Hierarchical crossover is modeled as the one-point crossover with the probability distribution defined in such a way, that the crossing point will be located between components. The probability distribution is given by the formula:

$$\chi_i = \begin{cases} \frac{b}{M_A - 1} d & (\exists k \in (0, l] : i = 2^k - 1) \wedge (\exists u \in (0, M_A] : \mathbf{1}^T i = upq) \\ 1 - b - \frac{b}{M_A - 1} d & \text{if } i = \mathbf{0} \\ \mathbf{0} & \text{in the other case} \end{cases} \quad (7)$$

To allow creating of the universal hierarchy trees with the components situated arbitrarily in the class, we should perform the transposition of bits coding one component before crossover.

**Definition 11.** Let  $v \in U_m$  and let  $k_1, k_2 \in \{0, \dots, l - 1\}$  be defined as follows:

$$k_1 = \min\{a \in (0, \dots, l - 1) : v_a = 1\}$$

$$k_2 = \max\{a \in (0, \dots, l - 1) : v_a = 1\} \quad (8)$$

Let  $\alpha : \Omega \rightarrow U_m$  be a random variable and let the function  $ts : U_m \times U_m \rightarrow U_m$  be defined according to the rule:  $\forall 0 \leq w < l$

$$ts_w(x, v) = \begin{cases} x_{k_1+i} & (\text{for } w = k_2 + i; \forall 0 \leq i < pg; k_1 + i < l; k_2 + i < l) \\ x_{k_2+i} & (\text{for } w = k_1 + i; \forall 0 \leq i < pg; k_1 + i < l; k_2 + i < l) \\ x_w & \text{in the other case} \end{cases} \quad (9)$$

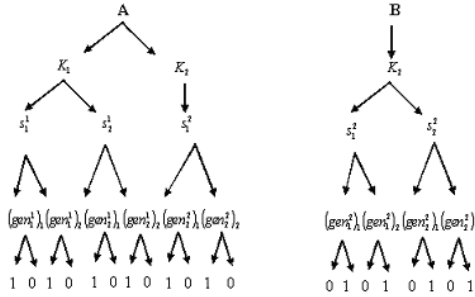
Transposition of components is the random operator defined as:  $ts(x, \alpha)$ . To make sure that only the bits from the component of the same class will be transposed, the probability distribution of chosen transposition mask is given by the formula:

$$P(\alpha = v) = \alpha_v =$$

$$\begin{cases} \binom{M_A/d}{2} & (\forall v : ((v, \mathbf{1}) = 2) \wedge (\lfloor \frac{k_1}{l/2} \rfloor = \lfloor \frac{k_2}{l/2} \rfloor) \wedge (k_1 \bmod pq = k_2 \bmod pq)) \\ 0 & \text{in the other case} \end{cases} \tag{10}$$

where  $d$  is the number of classes. From the construction of the binary strings for artifacts it follows, that the crossover operator with the distribution presented in equation (7) is the hierarchical crossover.

**Example 1.** Let  $P_A$  be the artifact space with all artifacts fulfilling the conditions: the number of components of arbitrary artifact  $B \in P_A$  is not bigger then  $M_A = 4$ , each component is described by  $p=2$  genes and each gene is coded by the binary string with the length  $q=2$ . Fig.3 presents exemplary hierarchy trees for artifacts from  $P_A$ . The universal hierarchy trees for hierarchy trees presented in fig.3 are shown on fig.4.



**Fig. 3.** Hierarchy trees for artifact from  $P_A$  - parents

The code of artifact A is the binary string  $x = 1010101010100000$ , and the code of artifact B is the binary string  $y = 0000000001010101$ . The crossover mask is chosen according to the equation (7). Let the crossover mask be as follows:  $z = 000000000001111$ . Let  $u$  equal 1. The result  $t$  of crossing is the following string:

$$t = K(x, y, z, u) = x \otimes z^u \oplus z^{1-u} \otimes y = 1010101010100101.$$

The universal hierarchy tree of the code  $t$ , presented on fig.5. consist of four components: first three components from individual A, and the last component from individual B. The same result will be obtained after hierarchical crossing with the crossover point (3, 1) with the same parents. To allow for creating

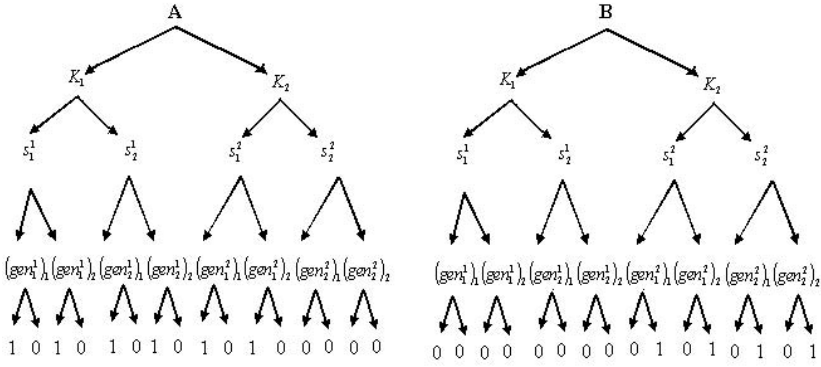


Fig. 4. Universal hierarchy trees for parents

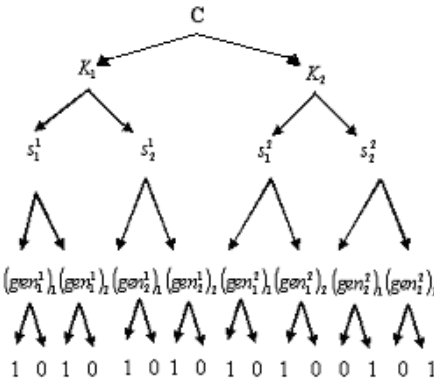


Fig. 5. The universal hierarchy tree for the results of crossing

of hierarchical trees with arbitrary located components of the same class, the transposition of components should be used before hierarchical crossover.

**Definition 12.** Let  $\kappa : \Omega \rightarrow 0, 1$  and  $\zeta, \eta, \alpha : \Omega \rightarrow U_m$  be random variables, and let the function  $M : U_m^6 \times 0, 1 \rightarrow U_m$  be defined according to the formula:

$$M(x, y, v, z, o_1, o_2, u) = mut(K(ts(x, o_1), ts(y, o_2), z, u)v) \tag{11}$$

Mixing is the random operator defined as:  $M(x, y, \eta, \zeta, \alpha_{o_1}, \alpha_{o_2}, \kappa)$ .

**Definition 13.** The heuristic search defined by mixing is the Markov chain  $\{\xi_i, i \in N\}$  with the transition matrix defined as follows:

$$Q_{j,i} = P(\xi_n = i | \xi_{n-1} = j) = r! \prod_{k=0}^{m-1} \frac{((G(j))_k)^{r_k}}{r_k!} \tag{12}$$

where  $G : \Omega \rightarrow A$  is given by the formula  $G_z(j) = M(x, y, \eta, \zeta, \alpha_{o_1}, \alpha_{o_2}, \kappa)$ .

**Lemma 1.**  $G(j) > 0$  for each  $z \in U_m$  and for each  $j \in \Omega_r$ .

**Conclusions.** The transition matrix  $Q_{j,i}$  has all terms positive. The markov chain describing the dynamics of the GA is ergodic. The probability, of obtaining an arbitrary state  $j$  starting from the arbitrary state  $j_0$  after finite number of epoch equals 1.

## 5 Conclusions

In this paper hierarchical chromosome based GA, which is used in design systems to solve complex design problems, as well as a Markov chain model for the algorithm, was presented. The model contains a method of coding of the artifacts, represented by various length trees, by means of fixed length binary strings and a method of the modeling of the hierarchical crossover. Presented formalization is used for finding the transition matrix, the investigation of ergodicity and asymptotic properties of genetic algorithms. It also allows us to proof that each population can be found with the probability equal to one within the finite number of steps. The presented formalization creates a base for research on ergodicity and asymptotic properties of various types of genetic algorithms based on hierarchical structures. The future work will involve theoretical investigation of the graph based genetic algorithms.

## References

1. Bentley P J (1997) Genetic Evolutionary Design of Solid Objects using a Genetic Algorithm. Ph.D. Dissertation, UCL London
2. Grabska E, Palacz W (2002) Hierarchical graphs in creative design. *Machine Graphics & Vision*, 9(1/2): 115-123
3. Paszyńska A (2005) An Extension of Vosešs Markov Chain Model for Genetic Algorithms. In *Proceedings of Genetic and Evolutionary Computation Conference*, Washington D. C.
4. Paszyńska A (2007) An application of hierarchical chromosome based genetic algorithm to the optimization of platform shape. In *Proceedings of X KAEiOG*, Bedlewo
5. Schaefer R (2002) *Podstawy genetycznej optymalizacji globalnej*. Wydawnictwo Uniwersytetu Jagiellonskiego, Kraków
6. Vose M D (1999) *The simple genetic algorithm*. MIT Press



---

# Binary Shape Characterization Using Morphological Boundary Class Distribution Functions

Marcin Iwanowski

Institute of Control and Industrial Electronics, Warsaw University of Technology,  
ul.Koszykowa 75, 00-662 Warszawa Poland  
iwanowski@isep.pw.edu.pl

**Summary.** In the paper a new method for binary shape characterization is proposed. It is based on the analysis of binary image pixels belonging to the internal boundary of an object performed by means of the mathematical morphology. By using morphological operators, internal boundary pixels are classified into three groups. This classification is performed for increasing sizes of structuring elements used. The changes within the class assignments are described by the boundary class distribution functions. These functions may be used as features in the pattern recognition process.

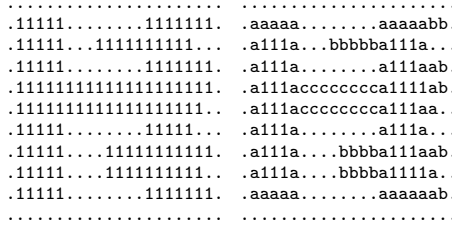
## 1 Introduction

In the process of binary shape recognition the shape features are widely used. In this paper new way of describing the binary shape is proposed. It is based on the mathematical morphology [7, 2]. By means of morphological operators, in the proposed method, pixels belonging to the internal object<sup>1</sup> boundary are classified into one of three spectral classes. The internal boundary is obtained here as the difference between image and its erosion. Every pixel belonging to the boundary is classified into one of the following classes:

1. Core boundary pixels - all pixels which have within their neighborhood at least one pixel of eroded object (referred to as *core region*). For regular shapes most of (or even all) boundary pixels belong to this group. For more complex shapes growing number of boundary pixels belong to the next two classes.
2. Corridors - pixels which are not core boundary pixels and which connect two core regions. As a result of the erosion a single object can be split into two or more objects (imagine e.g. eroded shape of a digit '8' with filled holes) The boundary pixels lying between these splitted areas are classified as corridors.
3. Branches - all the other boundary pixels. This class refers to all regions which do not connect multiple objects after erosion, but are too thin and too long to be classified as core boundary.

---

<sup>1</sup> Object is defined as a single connected component of pixels of value '1'.



**Fig. 1.** Test shape (left) and four types of boundary pixels (right): a - core boundary, b - branches, c - corridors

All these classes are shown on Fig. 1. The internal boundary in this example consists of all pixels marked as 'a', 'b' and 'c' and was computed using the structuring element containing all 8-neighbors and the pixel itself.

The proposed method classifies each internal boundary pixel to the appropriate class. The only parameter used in the method used is the structuring element of the morphological operators. Depending on this element, borders of various shapes and thicknesses can be obtained. To characterize the shape, the percentage of pixels belonging to a given class among all the pixels is computed. The proposed method is based on the assumption that the assignment of pixels to one of three classes depends also on the form of the structuring element used. For structuring elements of increasing sizes (i.e. covering pixel's neighborhood of increasing radius) the classification of boundary pixels changes differently for different shapes. Thus, the distribution of number of pixels belonging to each class for increasing sizes of the structuring element offers detailed shape description. The functions introduced in the paper are called *boundary class distribution functions*. They may be used as feature vector for shape description and recognition.

The paper is organized as follows. Section 2 contains the basic notions. Section 3 describes the pixel classification method. Section 4 introduces the boundary pixels distribution functions and discusses the choice of the structuring element. Section 5 presents some results. Finally, section 6 concludes the paper.

## 2 Basics

### 2.1 Internal Boundary of Binary Object

The internal boundary is defined as the set of pixels belonging to the object which have within their neighborhoods at least one pixel which does not belong to the object. In the proposed method this boundary is computed using the morphological gradient which is one of the most widely known morphological operators [7, 2]. In case of the binary images the word gradient is usually replaced by *contour*. In the proposed method, the morphological gradient (contour) by erosion has been applied. It is defined as:

$$\rho_B^-(X) = X \setminus (X \ominus B) \tag{1}$$

were  $X \ominus B$  stands for the erosion of binary image  $X$  with structuring element  $B$ . The contour line obtained by using the Eq. 1 is the internal contour of the object  $X$ . Usually the gradient is computed with the elementary or directional structuring element. In the first case the closest neighborhood in a given grid is considered. It results in either 4- or 8-pixel neighborhood. In case of the directional structuring element, the neighbor(s) located only in a given direction are taken into account. Apart from these traditional structuring elements also the *thick gradient* has also been defined in the literature [2]. In this case wider neighborhood is considered. The thickness and shape of these boundaries is defined by the size and shape of the structuring element. This feature of thick gradient makes it useful for shape analysis in the proposed method.

## 2.2 Morphological Reconstruction

Morphological reconstruction is the operation which allows extracting from the binary image only given objects (connected components). These objects are indicated using the supplementary image called *marker image*. Morphological reconstruction is defined as a series of geodesic (or conditional) dilations which are performed until idempotence of dilated image is reached. The morphological reconstruction (by dilation) is defined as:

$$R_Y(X) = \underbrace{(\dots((X \oplus B) \cap Y) \oplus B) \cap Y) \dots \cap Y}_{k\text{-times}} \quad (2)$$

where  $X \oplus B$  stands for the dilation of  $X$  with structuring element  $B$  and  $k$  is the lowest number of iterations after which the resulting image stops to change. Image  $X$  used is referred to as the *marker image* because it contains markers which indicated object on the second image -  $Y$ . The latter is called the *mask image*. The marker image  $X$  is reconstructed according to the content of the mask  $Y$ , which remains unchanged.

The morphological reconstruction can be computed either using the above definition or using specialized algorithms, which provides us with much faster computation of this operator [3].

## 2.3 Anchored Skeletonization

Skeletonization is used in the proposed methodology to find the corridor areas i.e. these areas that connect two core regions. The way of performing skeletonization is based on the notion of simple (or deletable) pixel [6] i.e. such a pixel that its removal does not change the homotopy of the binary image.

A foreground pixel  $p$  belonging to the image  $X$  is simple if and only if it satisfies the following three conditions: 1.  $\mathcal{N}_{\mathcal{G}}(p) \cap X \neq \emptyset$ , 2.  $\mathcal{N}_{\mathcal{G}'}(p) \cap X^C \neq \emptyset$ , 3.  $\exists S \in \mathcal{CC}_{\mathcal{G}}(\mathcal{N}_{\mathcal{G}}(p) \cap X)$  such that  $\mathcal{N}_{\mathcal{G}}(p) \cap X \subseteq S$ , where  $X^C$  stands for the complement of image  $X$ .  $\mathcal{N}_{\mathcal{G}}$  and  $\mathcal{N}_{\mathcal{G}'}$  represents the closest neighborhood of foreground and background pixels respectively. Due to connectivity paradox,

different connectivity should be used for the foreground and for the background. So either  $\mathcal{N}_{\mathcal{G}} = \mathcal{N}_8$  represents 8 closest neighbors (horizontal, vertical and diagonal one) and  $\mathcal{N}_{\mathcal{G}'} = \mathcal{N}_4$  represents 4 closest neighbors (without diagonal one), or inversely. Function  $CC_{\mathcal{G}}$  used in the above equation returns the set of  $\mathcal{G}$ -connected components of its argument. The fast computation of simpleness can be achieved using the look-up table for the neighborhood configurations.

By the successive removal of simple pixels the image is thinned. Image obtained by this thinning performed till idempotence is a *skeleton* which is also referred to as *homotopic marking*. In order to get symmetrical thinning, usually the operation is performed in two stage iterative process. In the stage phase the simple pixels are detected within the whole image (but not yet removed). The removal is performed in the second stage. These two stages are performed iteratively until idempotence is reached.

In order to have the possibility to control the thinning process, the notion of *anchor pixels* has been introduced [5]. These pixels are defined separately and - by definition - cannot be removed during the thinning process even if they are simple. The anchored skeletonization requires thus two input images - the image to be thinned and the image containing anchor pixels referred to as the *anchor image*.

### 3 Boundary Pixel Classification

The pixels belonging to the internal contour defined by the Eq. 1 are classified into three spatial classes [1]: core boundary, isolated region, corridor and branch. First, by using the morphological erosion with the given structuring element, the image object is shrinking. The shrunk areas of object will be referred to as *core regions*. The difference between the initial object and core area is the internal boundary of the object. Boundary pixels are further classified into three groups. The first of them is the locus of all pixels that are neighbors of the core region. Such pixels are classified as *core boundary pixels*. In fact core boundary pixels can be detected as the difference of input image opening (i.e. erosion followed by dilation) and erosion. It can be defined as the following function:

$$f_{cb}(X, B) = ((X \ominus B) \oplus B^T) \setminus (X \ominus B) \quad (3)$$

where  $X$  stands for initial image, and  $B^T = \{-p : p \in B\}$  for the transposition of structuring element  $B$ .  $f_{cb}(X, B)$  stands for the function which transforms binary image  $X$  to the image containing core boundary pixels using the structuring element  $B$ .

By subtracting core boundary from from the internal contour, the remaining part of the contour (rest of the boundary) is obtained:

$$f_{rob}(X, B) = X \setminus ((X \ominus B) \cup f_{cb}(X, B)) \quad (4)$$

The pixels belonging to this remaining part are further classified into two classes. To the first class belongs all regions that are stretched between at least

two core areas. They are *corridors* joining them. Pixels that are not corridors are simply *branches* of the core regions.

In order to find corridors, the anchored skeletonization is applied. The motivation for applying this transform was the fact that, when performing the anchored skeletonization by removal of simple pixels of the input image with the core areas used as anchors, the parts of input image that connects core region are detected. Since however, the result of skeletonization is one-pixel thick line, it does not cover the whole corridor areas. In order to the complete areas, the reconstruction is used. The whole operation can be expressed by the following expression:

$$f_{co}(X, B) = R_{f_{rob}(X, B)}(Skel(X, (X \ominus B)) \cap f_{rob}(X, B)) \quad (5)$$

where  $Skel(X_1, X_2)$  stands for the anchored skeletonization of the image  $X_1$  with anchor image  $X_2$ , and  $R$  represents the reconstruction operator according to the Eq. 2.

Using the equation 5, the second class - corridors - is detected. The remaining part of internal contour are the branches - the third class:

$$f_{br}(X, B) = X \setminus ((X \ominus B) \cup f_{cb}(X, B) \cup f_{co}(X, B)). \quad (6)$$

## 4 Boundary Pixel Classes Distribution Functions

The number of pixels belonging to each class depends, first of all, on the shape of the object being analyzed. But is also strongly depends on the structuring element  $B$  used - on its shape and size. Since the method is based on the classification of the internal boundary pixels, the way of obtaining this boundary (contour) is of primary importance. Usual way of computing the boundary makes use of 4- or 8-connected elementary structuring element, which consists of a central pixel plus 4 (horizontal and vertical) or 8 (horizontal, vertical and diagonal) closest neighbors respectively. To get more information on the shape the thick gradient approach [2] is more convenient in this case. Structuring element in this approach consist of neighboring pixels covering wider pixel neighborhood, usually a neighborhood of a given radius. Let then  $B^{(n)}$  be the neighborhood of a radius  $n$  i.e. set of the all the closest (to the central point) pixels which are not farther than  $n$ . The structuring element  $B^{(n)}$  can be defined in various ways. The simplest (and fastest) is based on superposition by successive dilations of  $n$  elementary structuring elements  $B^{(n)} = B' \oplus B' \oplus \dots \oplus B'$  where  $B'$  stands for an elementary structuring element. Since  $B'$  can be either 4- or 8-connected, the final shape of  $B^{(n)}$  can vary depending on it. The distance formulation which allows defining the radius is different for both cases. In the case of 4-connected  $B'$ , the  $B^{(n)}$  contains the pixels which are within the circle of radius  $n$  defined on the basis of the *city-block* distance. In the case of 8-connected  $B'$  the distance is implied by the *max-norm*. In order to get the neighborhood of radius  $n$  according to the *Euclidean* distance the superposition by dilations cannot be used and the structuring element  $B^{(n)}$  have to be computed individually for every  $n$ . This yields to slower computation but, on the other hand, gives the neighborhood closer to usual way of considering (which is based on Euclidean distance).

The classification of boundary pixels using the series of structuring elements  $B^{(n)}$  for increasing  $n$  allows obtaining the *boundary class distribution functions* for each boundary class. They are defined as:

$$\mathcal{D}(s) = |f(X, B^{(s)})| \tag{7}$$

where  $|\cdot|$  stands for the number of pixels of the argument. There are three functions defined according to the Eq. 7: *core boundary distribution*  $\mathcal{D}_{cb}$  for  $f \equiv f_{cb}$ , where  $f_{cb}$  is computed using the Eq. 3; *corridors distribution*  $\mathcal{D}_{co}$  for  $f \equiv f_{co}$  (Eq. 5) and *branches distribution*  $\mathcal{D}_{br}$  for  $f \equiv f_{br}$  (Eq. 6).

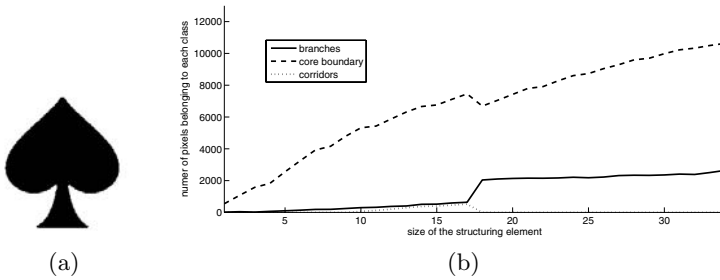
Alternatively the distribution functions can be defined using the relative number of pixels of each class, i.e. the ratio of the pixels of given class to the total number of pixels of the internal boundary:

$$\mathcal{D}'(s) = \frac{|f(X, B^{(s)})|}{|\rho_{B^{(s)}}^-(X)|} \tag{8}$$

where  $\rho^-$  stand for the internal boundary according to the Eq. 1.

## 5 Results

Figures 2 and 3 show two test figures: 'spades' and 'club' and their class distribution functions computed according to the Eq. 7. The classification was computed using the structuring element which approximated a disc of a given radius defined by Euclidean distance. The radii from 1 to 34 was taken into account. The distribution functions, especially branch and corridor ones are visibly different for both shapes. Thanks to that, they may be used for a shape classification.



**Fig. 2.** Test shape “spades” (a) and its distribution functions (b)

For some shapes only one function is sufficient to differentiate the shapes. In Fig. 4 the branch class distribution functions of another four test shapes are shown. Contrary to the previous example, they was computed according to the Eq. 8 and presents the ratio of the pixels belonging to the 'branch' class to the number of all internal boundary pixels. Similiarly to the previous example the structuring covering the neighborhood of a given radius according the

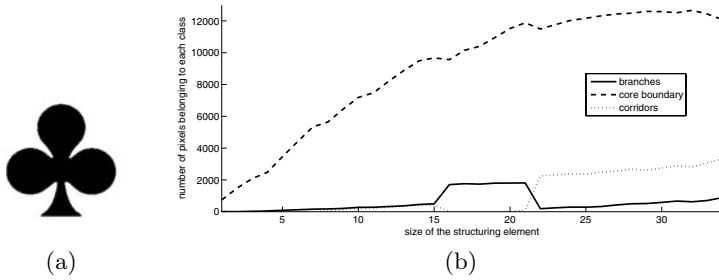


Fig. 3. Test shape “club” (a) and its distribution functions (b)

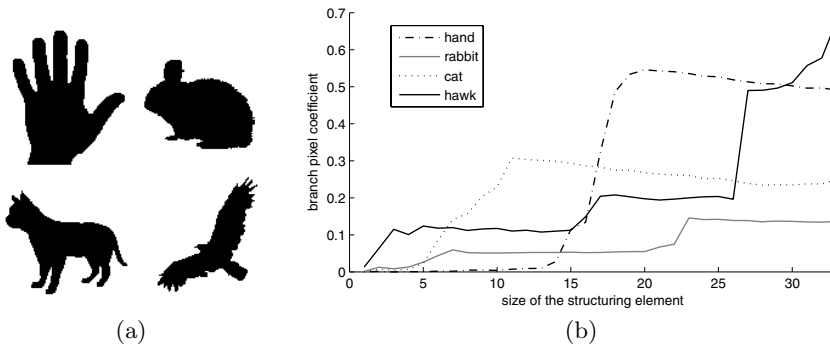


Fig. 4. Four test shapes (a) and their *branch* class distribution functions (b)

Euclidean distance was applied. Distribution functions for each shape varies from one shape to the other, what makes them a good feature for shape recognition.

## 6 Conclusions

In the paper the morphological method for binary shape description was proposed. It is based on the spatial classification of the boundary pixels into three groups: core boundary, corridors and branches. The classification method is based on the morphological operators of erosion, dilation and reconstruction as well as the anchored skeletonization. The parameter of the classifier is the structuring element used. Depending on the choice of this element various classification results are obtained. By computing the number of pixels belonging to each class as a function of the size of a structuring element, the boundary pixel class distribution functions are obtained. These functions characterize the shape and allow differentiating various shapes, what was also shown in the paper on some examples. Owing to that class distribution functions may be used as features for shape representation and pattern recognition.

## References

1. M.Iwanowski, "Morphological Boundary Pixel Classification", Prof. of IEEE EURO-CON Conference, Warsaw 2007.
2. P.Soille, "Morphological image analysis", *Springer Verlag*, 1999, 2004.
3. L.Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms", *IEEE Trans. on Image Processing*, Vol.2, No.2, April 1993.
4. J.Serra, L.Vincent, "An overview of morphological filtering", *Circuit systems Signal Processing*, 11(1), 1992.
5. L. Vincent, "Efficient computation of various types of skeletons", In: Loew, M., ed. *Medical Imaging V: Image Processing. Volume SPIE-1445 (1991)* pp.297-311
6. T.Kong, A.Rosenfeld, "Digital topology: Introduction and survey", *Computer Vision, Graphics, and Image Processing*, 1989, vol.48, pp.357-393
7. J.Serra, "Image analysis and mathematical morphology, vol.1", *Academic Press*, 1983.



---

# Fuzzy-aided Parsing for Pattern Recognition

Marzena Bielecka<sup>1</sup> and Marek Skomorowski<sup>2</sup>

<sup>1</sup> Department of Geoinformatics and Applied Computer Science, Faculty of Geology, Geophysics and Environmental Protection, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland

[bielecka@agh.edu.pl](mailto:bielecka@agh.edu.pl)

<sup>2</sup> Institute of Computer Science, Jagiellonian University, Nawojki 11, 30-072 Kraków, Poland

[skomorowski@ii.uj.edu.pl](mailto:skomorowski@ii.uj.edu.pl)

**Summary.** In syntactic pattern recognition a pattern is represented by abstract data, for instance a graph. The problem of recognition is to determine if a pattern, represented by the describing graph, belongs to a language  $L(G)$ , generated by a graph grammar  $G$ . The so-called *IE* graphs can be used for a pattern description. They are generated by so-called *ETPL(k)* graph grammars. The purpose of this paper is to present an idea of a new approach to syntactic recognition of fuzzy patterns represented by fuzzy *IE* graphs, followed the example of random *IE* graphs. This methodology can be used for analysis of wider class of patterns and scenes than the one described by the classical syntactic methods. In this paper, apart from the presentation of the fuzzy-aided approach, it is also shown that the probabilistic-syntactic approach is a special case of the presented one.

## 1 Introduction

The fundamental idea in syntactic pattern recognition is using of symbolic data like strings, trees and graphs for representation of a class of recognized objects ([3], [7], [9], [10]). The general scheme of syntactic pattern recognition and a scene analysis is following [7]. After pre-processing the recognized object is segmented in order to recognize the primitives the pattern consists of and relations between them. Decision, whether the analyzed pattern representation belongs to the class of objects describing by a given grammar, is made basing on the parsing algorithm.

In a node replacement graph grammar, a node of a derived graph is replaced by a new subgraph, which is connected to the remainder of the graph. A node replacement is controlled by a production of a given graph grammar. An example of a node replacement graph grammar is an *ETPL(k)* (*embedding transformation-preserving production-ordered k-left nodes unambiguous*) grammar introduced in [5]. The so-called *IE* (*indexed edge-unambiguous*) graphs have been defined in [5] for a description of patterns (scenes) in syntactic pattern recognition. Nodes in an *IE* graph denote pattern primitives. Edges between two nodes in an *IE* graph represent spatial relations between pattern primitives. An idea of a probabilistic improvement of syntactic recognition of distorted patterns represented

by graphs is described in [6] and [16]. A *random IE* graph approach ([6], [16]) is proposed for such a description and an efficient parsing algorithm for *IE* graphs (the computational complexity is  $O(n^2)$ ) is presented in [16].

The purpose of this paper is to present an idea of approach to syntactic recognition of fuzzy patterns represented by fuzzy *IE* graphs, followed the example of random *IE* graphs used for distorted pattern description. It turns out that, in a way, the fuzzy approach is a generalization of the probabilistic one. Fuzziness allows us not only described distortions in analyzed patterns but also give us possibility to describe in proper way patterns that can not be presented univocally. Furthermore there are a wide class of problems in which objects and/or spatial relations are described by fuzzy sets.

## 2 Motivation

In this section a few examples, in which the fuzzy-syntactic approach seems to be natural, are presented.

**First example.** Assume that during a manufacturing process a robotic inspection system checks type of a hole in a making elements, for instance plates, and spatial relations between holes. Assume also that there are a few standard types of holes and circular and quadratic ones are among them. Let, furthermore, the inspection system be based on a syntactic pattern recognition approach in which the holes are represented by nodes of graphs and spatial relations between holes by graph edges. A quadratic hole with rounded vertices can be regarded as a fuzzy object with partial membership to classes of both circular and quadratic holes. In this example nodes description as fuzzy sets is a natural approach. In this case membership functions describing fuzzy sets can be define basing on axiomatic method ([1]).

**Second example.** Considering the previous example assume that robotic inspection of technological process is based on statistical distribution of inaccuracy frequencies ([6], [12]). If a hole is made in a sufficient accuracy it is accept by the system. Not only the hole shape but also its location should be taken into consideration. Since inaccuracies of holes location influence each other, the simple statistical analysis can be insufficient to make a decision. In such a case a fuzzy inference can be applied. Then, holes and their locations can be represented by fuzzy sets and membership functions can be calculated using the statistical distribution according to the methodology described in ([1]). Let, like in the first example, the inspection system is based on a syntactic pattern recognition in which the holes are represented by nodes of graphs and spatial relations between holes by graph edges. In this example both the graph nodes and its edges would be described as fuzzy sets. Automatic focusing vision system for inspection of size and shape and positions of small holes in the context of precision engineering, described in ([8]), is an example of a system performing such type of task.

**Third example.** Consider an autonomous mobile agent. Assume that it has to navigate in an unchanging environment. A helicopter flying autonomously in a

textured urban environment is an example of such agent ([11]). As it has been already mentioned it should be equipped with symbolic representation of the surrounding world in order to analyze the scene they act on ([4], [15]). Let its vision system be a syntactic one based on graph representation of the spatial relationships between obstacles the agent should navigate among. Let according to, for instance, the optimization requirements, the system prefers one direction but admits also another ones allowing to navigate without collision. In such a case the scene would be represented by a classical (i.e. not fuzzy) IE graph but directions the agent can choice would be represented by fuzzy sets. The decision making system would be based on fuzzy inference.

**Fourth example.** Let us consider computer-aided analysis and recognition of pathological wrist bone lesions ([13], [17]). This method consists on analysis of the structure of the said bones based on palm radiological images. During pre-processing operations in the examined X-ray images the bones contours were separated and a graph, representing bones and spatial relation between them, was spanned. In the beginning, spatial relationships given by the graph edges were represented by single directions ([17]) but later each basic spatial relationship was represented as angular interval ([13]). The second approach can be interpreted in such a way that every basic spatial relationship is described as a fuzzy set for which its membership function has positive values on the specified angular interval and is equal to zero outside this interval. It should be mentioned that in ([13]) such interpretation was not considered.

Recapitulating, four examples in which various aspects of possibility of improve syntactic approach by fuzzy sets has been discussed. The classical IE graphs can be generalized by including fuzzy sets to description their nodes (example 1), edges (example 4), both the nodes and edges (example 2) and fuzzy inference approach can be applied to classical graphs (i.e. non-fuzzy ones) - example 3.

### 3 Fuzzy *IE* Graphs for Fuzzy Patterns Representation

Let us remind definition of an *IE* graph ([5]). An indexed edge-unambiguous graph, an *IE* graph over  $\Sigma$  and  $\Gamma$  is a quintuple

$$g = (V, E, \Sigma, \Gamma, \Phi),$$

where

$V$  - is a finite, nonempty set of nodes to which indices have been ascribed in an unambiguous way,

$\Sigma$  - is a finite, nonempty set of node labels,

$\Gamma$  - is a finite, nonempty set of edge labels,

$E$  - is a set of edges of the form  $(v, \lambda, w)$ ,  $v, w \in V$ ,  $\lambda \in \Gamma$ , such that index of  $v$  is less than index of  $w$ ,

$\Phi : V \rightarrow \Sigma$  - is a node labelling function.

Assume that both labelled objects in nodes of a graph and spatial relations are represented by fuzzy sets of a first order with membership functions  $\mu_i$  and  $\nu_i$  respectively. Let, furthermore, the set of all objects  $\Sigma$  be  $n$ -elemental and the set of all spatial relations be  $k$ -elemental. Let us define, informally, a *fuzzy IE* graph as an *IE* graph in which nodes labels are replaced by a vector  $\mu = [\mu_1, \dots, \mu_n]$  of values of membership functions  $\mu_i, i \in \{1, \dots, n\}$  and edges labels are replaced by vector  $\nu = [\nu_1, \dots, \nu_k]$  of values of membership functions  $\nu_j, j \in \{1, \dots, k\}$  - see Fig.1.

The fuzzy measure of an outcome *IE* graph, obtained form a given *fuzzy IE* graph, is equal to the value of a  $T$ -norm  $T$  of the values components of the node and edge vectors. An axiomatic definition of  $T$ -norms is given in [14], definition 4.22, page 80. Having a *fuzzy IE* graph  $R$  the fuzzy measure  $\lambda_r$  of an outcome graph  $r$  is calculated as

$$\lambda_r = T \left( \prod_{p=1}^P \left( \mu_{f_r(p)}^p \right), \prod_{s=1}^S \left( \nu_{g_r(s)}^s \right) \right)$$

where

$p$  - is a number of a regarded node,

$s$  - is a number of an edge,

$f_r(p)$  - is a chosen component number of a vector  $\mu^p$  whereas

$g_r(s)$  - is a chosen number of component of a vector  $\nu^s$ .

If the arithmetic product is used as a  $T$ -norm then the presented parsing algorithm (see Section 3) is identical to the random parsing algorithm described in [16].

## 4 Parallel Parsing of Fuzzy *IE* Graphs

Given an unknown pattern represented by a *fuzzy IE* graph  $R$ , the problem of recognition of a pattern under study is to determine if an outcome *IE* graph  $r$ , obtained from the *fuzzy IE* graph  $R$ , belongs to a graph language  $L(G)$  generated by an *ETPL*( $k$ ) graph grammar  $G$ . In the proposed parallel and cut-off strategy of *fuzzy IE* graph parsing a number of simultaneously derived graphs is equal to a certain number *limit*. In this case, derived graphs spread through the search tree, but only the best, that is with maximum measure value, *limit* graphs are expanded.

Let us introduce the following notations:  $Z$  - a starting graph of an *ETPL*( $k$ ) graph grammar  $G$ ,  $P$  - is a set of productions of the grammar  $G$ ,  $R$  - an analyzed fuzzy graph. The idea of the proposed parsing algorithm is the following:

1. Apply to the starting graph  $Z$  productions belonging to the set  $P$ , which are admissible for further derivations. If such productions do not exist then stop the parsing algorithm.
2. For subgraphs obtained in the point 1, compute values of their membership functions using a given  $T$ -norm.

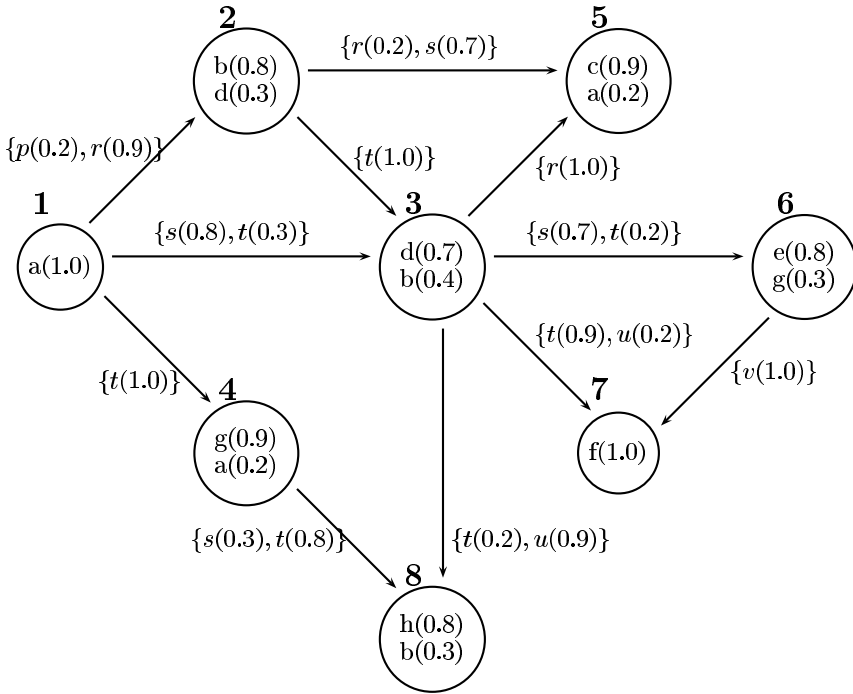


Fig. 1. An example of fuzzy *IE* graph representing an unknown pattern

3. For further derivation choose a number (*limit*) of derived subgraphs with the biggest values of membership function.
4. To the chosen subgraphs apply these productions from *P* which are admissible for further derivations. If such productions do not exist then stop the parsing algorithm.
5. Repeat the points 3 and 4 until fuzzy-outcome graph *R* is obtained.
6. Stop the parsing algorithm.

An example of the introduced algorithm application can be found in [2].

## References

1. Bielecka M (2006) A method of membership function construction in fuzzy systems. In: Cader A, Rutkowski L, Tadeusiewicz R, Żurada J (eds) Challenging Problems of Science - Computer Science, Academic Publishing House EXIT, Warszawa, 111-117
2. Bielecka M, Skomorowski M, Bielecki A (2007) Fuzzy-syntactic approach to pattern recognition and scene analysis. Proceedings of the 4th International Conference on Informatics in Control, Automatics and Robotics ICINCO'07, ICSO - Intelligent Control Systems and Optimization, Robotics and Automation, vol.1, 29-35

3. Chen SL, Chen Z, Li RK (1991) A DGR method for extracting the topology of an upper-half profile of a turned part from CAD data. *Computer Integrated Manufacturing* 4: 45-56
4. Ferber J (1999) *Multi-Agent Systems. An Introducing to Distributed Artificial Intelligence*. Addison-Wesley, Harlow
5. Flasiński M (1993) On the parsing of deterministic graph languages for syntactic pattern recognition. *Pattern Recognition* 26: 1-16
6. Flasiński M, Skomorowski, M (1998) Parsing of random graph languages for automated inspection in statistical-based quality assurance systems. *Machine Graphics and Vision* 7: 565-623
7. Fu KS (1982) *Syntactic Pattern recognition and Application*. Prentice Hall, New York
8. Han M, Han H (1999) Automating focusing vision system for inspection of size and shape of small holes. *Journal of the Korean Society of Precision Engineering* 16: 80-86
9. Jakubowski R (1997) Structural approach to modelling and analysis of 2D-shapes. *Bull. Polish Academy of Sciences - Technical Sciences* 45: 373-387
10. Jakubowski R, Stapor K (1999) Structural method in perceptual organization of 2D-curve detection. *Bull. Polish Academy of Sciences - Technical Sciences* 47: 309-324
11. Muratet L, Doncieux S, Briere Y, Meyer JA (2005) A contribution to vision-based autonomous helicopter flight in urban environment. *Robotics and Autonomous Systems* 50: 195-229
12. Noori H., Radford R (1995) *Production and Operation Management - Total Quality and Responsiveness*. McGraw Hill
13. Ogiela M, Tadeusiewicz R, Ogiela L (2006) Image languages in intelligent radiological palm diagnosis. *Pattern Recognition* 39: 2157-2165
14. Rutkowski L (2005) *Artificial Intelligence Techniques and Methods*. PWN, Warszawa (in Polish)
15. Scheier C, Pfeifer R (1999) The embodied cognitive science approach. In: Tschacher W, Dauwalde J.P. (eds) *Dynamics, Synergetics, Autonomous Agents, Studies of Nonlinear Phenomena and Life Science*, vol.8, World Scientific, Singapore, 159-179
16. Skomorowski, M (1999) Use of random graph parsing for scene labeling by probabilistic relaxation. *Pattern Recognition Letters* 20: 949-956
17. Tadeusiewicz R, Ogiela M (2005) Picture languages in automatic radiological palm interpretation. *Int. J. Appl. Mathematics and Computer Science* 15: 305-312

---

# On Speeding Up the Learning Process of Neuro-fuzzy Ensembles Generated by the Adaboost Algorithm

Marcin Korytkowski<sup>1,2</sup>, Leszek Rutkowski<sup>1</sup>, and Rafał Scherer<sup>1</sup>

<sup>1</sup> Department of Computer Engineering, Częstochowa University of Technology  
al. Armii Krajowej 36, 42-200 Częstochowa, Poland

<sup>2</sup> Olsztyn Academy of Computer Science and Management  
ul. Artyleryjska 3c, 10-165 Olsztyn, Poland

marcink@kik.pcz.czest.pl, lrutko@kik.pcz.czest.pl, rafal@ieee.org

**Summary.** Boosting is one of the most popular methods of multiple classification. In the paper we propose a method for speeding up the learning process by modifying the backpropagation algorithm and fuzzy clustering algorithm for boosting learning of several neuro-fuzzy classifiers. Simulations show superior performance of our method comparing to a standard boosting classification.

## 1 Introduction

Classifiers can be combined to improve accuracy. By combining intelligent learning systems, the model robustness and accuracy is nearly always improved, comparing to single-model solutions. Popular methods are bagging and boosting which are meta-algorithms for learning different classifiers. They assign weights to learning samples according to their performance on earlier classifiers in the ensemble. Thus subsystems are trained with different datasets. In the paper we develop a method for speeding up the gradient learning process in neuro-fuzzy system ensembles by modifying the backpropagation algorithm and the FCM clustering algorithm. In the next section we describe the AdaBoost algorithm, in Section 3 we propose a new method for backpropagation learning and in Section 4 we modify the FCM algorithm for boosting. We verify the methods on a well known data benchmark in Section 5.

## 2 Adaboost Algorithm

In this section we describe the AdaBoost algorithm which is the most popular boosting method [4],[7],[9]. Let us denote the  $l$ -th learning vector by  $\mathbf{z}^l = [x_1^l, \dots, x_n^l, y^l]$ ,  $l = 1 \dots m$  is the number of a vector in the learning sequence,  $n$  is a dimension of input vector  $\mathbf{x}^l$ , and  $y^l$  is the learning class label. Weights, assigned to learning vectors, have to fulfill the following conditions

$$(i) 0 < D^l < 1, \quad (ii) \sum_{l=1}^m D^l = 1. \quad (1)$$

The weight  $D^l$  is the information how well classifiers learned in consecutive steps of an algorithm for a given input vector  $\mathbf{x}^l$ . Vector  $\mathbf{D}$  for all input vectors is initialized according to the following equation

$$D_t^l = \frac{1}{m}, \quad \text{for } t = 0, \quad (2)$$

where  $t$  is the number of a boosting iteration (and a number of a classifier in the ensemble). Let  $\{h_t(\mathbf{x}) : t = 1, \dots, T\}$  denotes a set of hypotheses obtained in consecutive steps  $t$  of the algorithm being described. For simplicity we limit our problem to a binary classification (dichotomy). Similarly to learning vectors weights, we assign a weight  $c_t$  for every hypothesis, such that

$$(i) \sum_{t=1}^T c_t = 1, \quad (ii) c_t > 0. \quad (3)$$

Now in the AdaBoost algorithm we repeat steps 1-4 for  $t = 1, \dots, T$ :

1. Create hypothesis  $h_t$  and train it with a data set with respect to a distribution  $d_t$  for input vectors.
2. Compute the classification error  $\varepsilon_t$  of a trained classifier  $h_t$  according to the formula

$$\varepsilon_t = \sum_{l=1}^m D_t^l(z^l) I(h_t(\mathbf{x}^l) \neq y^l), \quad (4)$$

where  $I$  is the indicator function

$$I(a \neq b) = \begin{cases} 1 & \text{if } a \neq b \\ 0 & \text{if } a = b \end{cases}. \quad (5)$$

If  $\varepsilon_t = 0$  or  $\varepsilon_t \geq 0.5$ , stop the algorithm.

3. Compute the value

$$\alpha_t = 0.5 \ln \frac{1 - \varepsilon_t}{\varepsilon_t}. \quad (6)$$

4. Modify weights for learning vectors according to the formula

$$D_{t+1}(\mathbf{z}^l) = \frac{D_t(\mathbf{z}^l) \exp\{-\alpha_t \mathbf{I}(h_t(\mathbf{x}_l) = y^l)\}}{N_t}, \quad (7)$$

where  $N_t$  is a constant such that  $\sum_{l=1}^m D_{t+1}(\mathbf{z}^l) = 1$ . To compute the overall output of the ensemble of classifiers trained by AdaBoost algorithm the following formula is used

$$f(\mathbf{x}) = \sum_{t=1}^T c_t h_t(\mathbf{x}), \quad (8)$$



where

$$c_t = \frac{\alpha_t}{\sum_{t=1}^T |\alpha_t|} \tag{9}$$

is classifier importance for a given training set. The AdaBoost algorithm is a meta-learning algorithm and does not determine the way of learning for classifiers in the ensemble.

### 3 Backpropagation Learning in Boosting Neuro-fuzzy Ensembles

In this section we show a modification of the backpropagation algorithm for speeding up learning in boosting ensembles. In the first step of the above algorithm there exists imprecise statement "train it with a data set with respect to the distribution for input vectors  $\mathbf{x}$ ". There exists several methods for taking into account weights  $D_t^q$ ,  $t = 1, \dots, T$ ,  $q = 1, \dots, m$  in the process of learning. Please note that  $q = l$  for the first epoch. There are several methods to take into account the input vector weights during training. Let us assume that our classifiers are neural networks or neuro-fuzzy systems trained by the backpropagation algorithm. One of the methods is based on picking samples at random with the probability  $D_t^q$  being equal to weights of samples  $\mathbf{x}^q$ ,  $q = 1, \dots, m$  and in this way creating a new learning set for the hypothesis  $h_t$ . Unfortunately this method is nondeterministic and researchers obtain different results using it. In [4] it is recommended to use, for each hypothesis, modified error function which depends on weight values for learning vectors. The basis for further work can be a formula given in [7], where standard error in the backpropagation algorithm is multiplied by sample weights

$$Q^q(l) = (y^q(l) - h_t^q(\mathbf{x}^q(l)))^2 D_t^q(l), \quad q = 1, \dots, M, \quad t = 1, \dots, T. \tag{10}$$

This approach has the following drawback. Let us assume that in a learning set there is a large number of vectors, say 10 000. From (10) it results that weight values in initial steps of AdaBoost will be very small, e.g.  $d_t^q = \frac{1}{10000}$  for  $t = 1$ . Error values used during training are very small, so the training will be quite long. Please note that the weights are modified after a given classifier is trained. To overcome the problem a new method for incorporating sample weights will be proposed.

In the paper we use Mamdani neuro-fuzzy systems designed using the Larsen rule. Output of such systems is given by

$$y = \frac{\sum_{r=1}^N \bar{y}^r \cdot \mu_{\bar{B}^r}(\bar{y}^r)}{\sum_{r=1}^N \mu_{\bar{B}^r}(\bar{y}^r)}, \tag{11}$$

where membership functions of fuzzy sets  $\bar{B}^r$ ,  $r = 1, \dots, N$ , are given by

$$\mu_{\bar{B}^r}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \left[ \mu_{A'}(x) \overset{T}{*} \mu_{A^r \rightarrow B^r}(x, y) \right]. \tag{12}$$

If we assume that we use singleton defuzzification then above formula takes the following form

$$\mu_{\bar{B}^r}(y) = \mu_{A^r \rightarrow B^r}(\bar{x}, y) = T(\mu_{A^r}(\bar{x}), \mu_{B^r}(y)). \tag{13}$$

Let us observe that

$$\mu_{A^r}(\bar{x}) = \overset{n}{T}_{i=1}(\mu_{A_i^r}(\bar{x}_i)), \tag{14}$$

and assume that

$$\mu_{B^r}(\bar{y}^r) = 1. \tag{15}$$

Then, using a t-norm property ( $T(a, 1) = a$ ), we obtain a formula describing a single Mamdani fuzzy system

$$\bar{y} = \frac{\sum_{r=1}^N \bar{y}^r \cdot \overset{n}{T}_{i=1}(\mu_{A_i^r}(\bar{x}_i))}{\sum_{r=1}^N \overset{n}{T}_{i=1}(\mu_{A_i^r}(\bar{x}_i))}. \tag{16}$$

Linguistic variables are described by the Gaussian membership function

$$\mu_{A_i^r}(x_i) = \exp \left[ - \left( \frac{x_i - \bar{x}_i^r}{\sigma_i^r} \right)^2 \right]. \tag{17}$$

If we choose product t-norm, then the neuro-fuzzy system with the center average defuzzification and Mamdani-type relation is described by

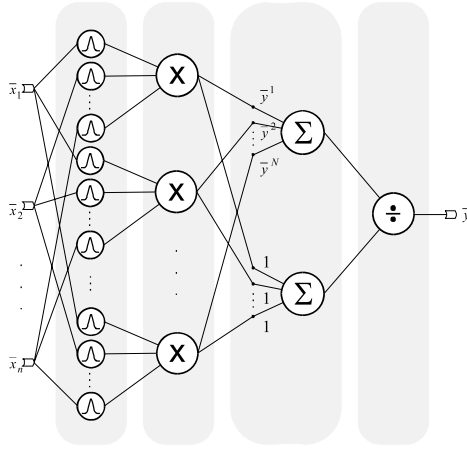
$$\bar{y} = \frac{\sum_{r=1}^N \bar{y}^r \left[ \prod_{i=1}^n \exp \left( - \left( \frac{x_i - \bar{x}_i^r}{\sigma_i^r} \right)^2 \right) \right]}{\sum_{r=1}^N \left[ \prod_{i=1}^n \exp \left( - \left( \frac{x_i - \bar{x}_i^r}{\sigma_i^r} \right)^2 \right) \right]}, \tag{18}$$

and the network structure of the system is depicted in Fig. 1. Let us denote a pair of input vector and output signal by  $\mathbf{x}^q$  and  $d^q$ , respectively, where  $q$  denotes vector number,  $q = 1, \dots, M$ . Let us define the error

$$Q^q = \frac{1}{2} [f(\mathbf{x}^q) - d^q]^2. \tag{19}$$

During learning we tune parameters  $\bar{y}^r, \bar{x}_i^r, \sigma_i^r$  to minimize (19) for all learning vectors. The most common approach is the use of gradient algorithms. Value of  $\bar{y}^r$  parameter is computed by

$$\bar{y}^r(l+1) = \bar{y}^r(l) - \eta \frac{\partial Q}{\partial \bar{y}^r} | l, \tag{20}$$



**Fig. 1.** Single Mamdani neuro-fuzzy system

where  $l = 0, 1, 2, \dots$  is a number of step of gradient algorithm, and  $\eta$  is a learning coefficient. Output signal value depends on  $\bar{y}^r$  only by the numerator of (18). Let us assume that  $a = \sum_{r=1}^N \bar{y}^r z^r$ ,  $b = \sum_{r=1}^N z^r$  and  $z^r = \prod_{i=1}^n \exp\left(-\left(\frac{x_i - \bar{x}_i^j}{\sigma_i^j}\right)^2\right)$ . Then we can write modification of the parameter  $\bar{y}^r$  for consecutive steps as follows

$$\bar{y}^r(l + 1) = \bar{y}^r(l) - \eta \frac{f - d}{b} z^r. \tag{21}$$

An analogical deliberation for parameters  $\bar{x}_i^r$  and  $\sigma_i^r$  yields

$$\bar{x}_i^r(l + 1) = \bar{x}_i^r(l) - \eta \frac{f - d}{b} (\bar{y}^r - f) z^r \frac{2(x_i^q - \bar{x}_i^r(l))}{\sigma_i^r{}^2(l)}, \tag{22}$$

$$\sigma_i^r(l + 1) = \sigma_i^r(l) - \eta \frac{f - d}{b} (\bar{y}^r - f) z^r \frac{2(x_i^q - \bar{x}_i^r(l))^2}{\sigma_i^r{}^3(l)}. \tag{23}$$

Formulas (21-23) can be treated as a special case of the backpropagation algorithm. If we would like to build a modular learning system based on boosting consisted of several neuro-fuzzy systems with gradient learning, we end up with similar problems as in modular boosting neural networks. Here we propose a method for neural networks [8] adopted for neuro-fuzzy systems. Learning coefficient for a sample  $q$  is computed in every step  $t$  of the algorithm. We tend to a situation where samples with maximal weight values have the same learning coefficient as set by the researcher, i.e.  $\nu_t^q = \eta$  for samples for which  $D_t^{\max} = \max_{i=1, \dots, M} D_t^i$  where  $m$  is the number of learning vectors. For the rest of samples the learning coefficient should be respectively smaller. To fulfill the

above assumptions we propose the following modification of the learning coefficient for all learning vectors

$$\nu_t^q = \eta \frac{D_t^q + 1}{D_t^{\max} + 1}, \quad (24)$$

where  $D_t^{\max}$  is the maximal value of sample weights for  $t$ -th hypothesis. Finally we obtain the new formulas for parameters update in an ensemble of Mamdani neuro-fuzzy systems

$$\begin{aligned} \bar{y}_t^r(l+1) &= \bar{y}_t^r(l) - \nu_t^q \frac{f_t-d}{b_t} z_t^r = \\ \bar{y}_t^r(l) - \left( \eta \frac{D_t^q+1}{D_t^{\max}+1} \right) \frac{f_t-d}{b_t} z_t^r, \end{aligned} \quad (25)$$

$$\begin{aligned} \bar{x}_i^{(r)t}(l+1) &= \bar{x}_i^{(r)t}(l) - \nu_t^q \frac{f_t-d}{b_t} (\bar{y}_t^r - f_t) z_t^r \frac{2(x_i^{(p)t} - \bar{x}_i^{(r)t}(l))}{\sigma_i^{(r)t2}(l)} = \\ \bar{x}_i^{(r)t}(l) - \left( \eta \frac{D_t^q+1}{D_t^{\max}+1} \right) \frac{f_t-d}{b_t} (\bar{y}_t^r - f_t) z_t^r \frac{2(x_i^{(p)t} - \bar{x}_i^{(r)t}(l))}{\sigma_i^{(r)t2}(l)}, \end{aligned} \quad (26)$$

$$\begin{aligned} \sigma_i^{(r)t}(l+1) &= \sigma_i^{(r)t}(l) - \nu_t^q \frac{f_t-d}{b_t} (\bar{y}_t^r - f_t) z_t^r \frac{2(x_i^{(p)t} - \bar{x}_i^{(r)t}(l))^2}{\sigma_i^{(r)t3}(l)} = \\ \sigma_i^{(r)t}(l) - \left( \eta \frac{D_t^q+1}{D_t^{\max}+1} \right) \frac{f_t-d}{b_t} (\bar{y}_t^r - f_t) z_t^r \frac{2(x_i^{(p)t} - \bar{x}_i^{(r)t}(l))^2}{\sigma_i^{(r)t3}(l)}, \end{aligned} \quad (27)$$

where  $d$  is desired output and  $f$  is the output of a neuro-fuzzy system. In the next section we propose a modification of the FCM clustering algorithm for boosting ensemble modification.

## 4 FCM Modification for Boosting Initialization

One of several advantages of fuzzy systems is the possibility of interpretation of the knowledge. Thanks to this we can also initialize the parameters of the fuzzy system. A common method for such initialization is the use of the Fuzzy  $C$ -Means algorithm [1], [2], [5]. The FCM algorithm is based on fuzzy clustering of data vectors and assigning membership values to them. The algorithm is derived by minimizing the criterion [6]

$$J(\mathbf{X}; \mathbf{U}; \mathbf{V}) = \sum_{i=1}^C \sum_{q=1}^M (\mu_{iq})^m \|\mathbf{x}_q - \mathbf{v}_i\|_{\mathbf{A}}^2 \quad (28)$$

and  $\mathbf{U} = [\mu_{iq}] \in Z$  is a partition matrix of dataset  $\mathbf{X}$ , and  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_C]$  is a vector of centers to be determined by the algorithm,  $\mathbf{v}_i \in R^n$ ,  $i = 1, \dots, C$ . The algorithm can be stopped when matrix  $\mathbf{U}$  does not change or the change is below certain level ( $\|\mathbf{U}^{(l+1)} - \mathbf{U}^{(l)}\| < \varepsilon$ ). Alternatively we can check the change of centers, i.e.  $\|\mathbf{V}^{(l+1)} - \mathbf{V}^{(l)}\| < \varepsilon$ , where  $l$  the iteration number in the FCM algorithm. We modify the FCM algorithm by adding sample weight  $D_t^q$  to the formula for computing data objects memberships, please see Fig.2.

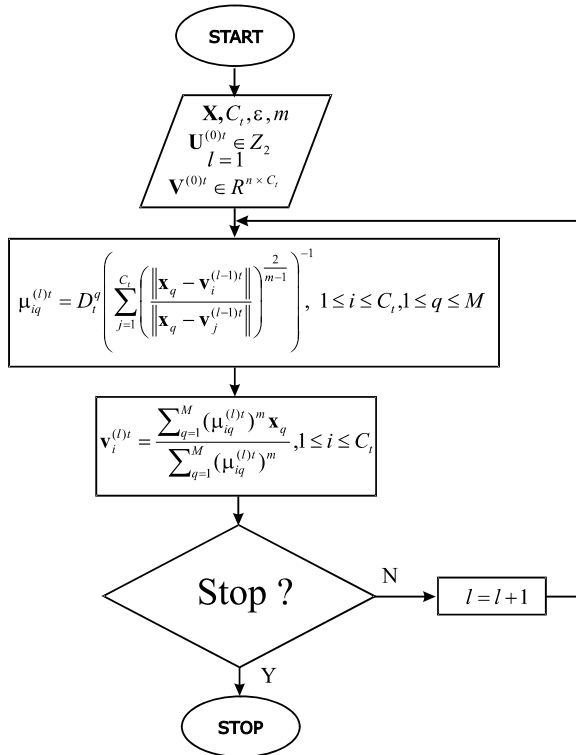


Fig. 2. Flowchart of the modified FCM algorithm

### 5 Numerical Simulations

In this section we test the efficiency of our methods on the Glass Identification problem [3]. The goal is to classify 214 instances of glass into window and non-window glass basing on 9 numeric features. We took out 43 instances for a testing set. We obtained 97.67% classification accuracy. We used 5 neuro-fuzzy classifiers described in Table 1. Simulations shows superior performance and faster convergence of the proposed approach comparing to the literature.

Table 1. Numerical results of Glass Identification

	Classifier 1	Classifier 2	Classifier 3	Classifier 4	Classifier 5
No. of rules	2	2	2	2	2
No. of epochs	20	30	50	100	40
Coefficients $c_t$	0.13	0.01	0.17	0.34	0.34
Classification %	88.37	76.74	90.6	97.67	97.67

## 6 Conclusions

The paper presents two methods improving the Adaboost algorithm. The first improvement is obtained by the novel method of incorporating sample weights into gradient learning. The second one is a modification of the FCM clustering algorithm to better initialize a neuro-fuzzy ensemble. Simulations shows superior performance and faster convergence of the proposed approach comparing to the literature.

**Acknowledgement.** This work was supported in part by the Foundation for Polish Science (Professorial Grant 2005-2008) and the Polish State Committee for Scientific Research (Grant Nr T11C 04827).

## References

1. Bezdek JC, Pal SK (1992) *Fuzzy Models for Pattern Recognition*, IEEE Press, New York
2. Bezdek J, Keller J, Krisnapuram R, Pal NR (1999) *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Kluwer Academic Press
3. Blake CL, Merz CJ (1998) UCI Repository of machine learning databases, [www.ics.uci.edu/~mllearn/MLRepository.html](http://www.ics.uci.edu/~mllearn/MLRepository.html), Irvine, University of California, Department of Information and Computer Science
4. Breiman L (1997) Bias, variance, and arcing classifiers. In: Technical Report 460, Statistics Department, University of California
5. Czogala E, Leski J (2000) *Fuzzy and Neuro Fuzzy Intelligent Systems*, Physica Verlag, Heidelberg, New York
6. Konar A (2005) *Computational Intelligence*, Springer, Berlin Heidelberg New York
7. Meir R, Ratsch G, (2003) *An Introduction to Boosting and Leveraging*, Advanced Lectures on Machine Learning
8. Korytkowski M, Rutkowski L, Scherer R (2006) On Combining Backpropagation with Boosting, 2006 International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence, Vancouver, BC, Canada
9. Schapire RE (1999) A brief introduction to boosting, Proc. of the Sixteenth International Joint Conference on Artificial Intelligence.

---

# Active Hypercontour as Information Fusion Method

Piotr S. Szczepaniak<sup>1,2</sup>

<sup>1</sup> Institute of Computer Science, Technical University of Lodz, Wolczanska 215, 90-924, Lodz, Poland

<sup>2</sup> Systems Research Institute, Polish Academy of Sciences, ul.Newelska 6, 01-447 Warsaw, Poland

**Summary.** In this paper, it is shown that active hypercontours can be interpreted as a method for information fusion. The fusion concerns using of information from diverse sources and of different type (data and knowledge). Moreover, active hypercontours can work as group classifiers.

## 1 Introduction

Roughly speaking *information fusion* is a methodological and computational approach which allows the use of different forms of information from diverse sources to solve the given task - Fig. 1. Application of number of systems concurrently solving the task is also recommended; here, a method for taking the final decision must be proposed [1-8] - a classical example is the majority voting. Another closely related problems concerns the criteria for choice of classifiers, development of metaclassifiers, sequential recognition, consideration of uncertain information, etc. [1-13]. A variety of problems can be stated, but we focus our attention on clustering and classification.

Let us remind definitions of basic concepts, namely data, information and knowledge:

- *Data* - Formalized representation of facts, numbers or notions that enables their manual as well as automatic transmission and processing. Numerical data are represented by characters and specialized symbols.
- *Information* - Meaning that is assigned to data in terms of a suitable convention proper for a given type of both data and processing system. This notion is also used when results of data processing are considered. Information scientists and practitioners often happen use the notions of data and information interchangeably.
- *Knowledge* - Awareness or familiarity gained by experience; a theoretical or practical understanding of a subject.

Originally, *active contour* methods [14, 15, 16, 21] were developed as tools for the low-level image segmentation (Fig. 2), but with the ability to use high-level information. The main idea of this approach is to find an optimal contour

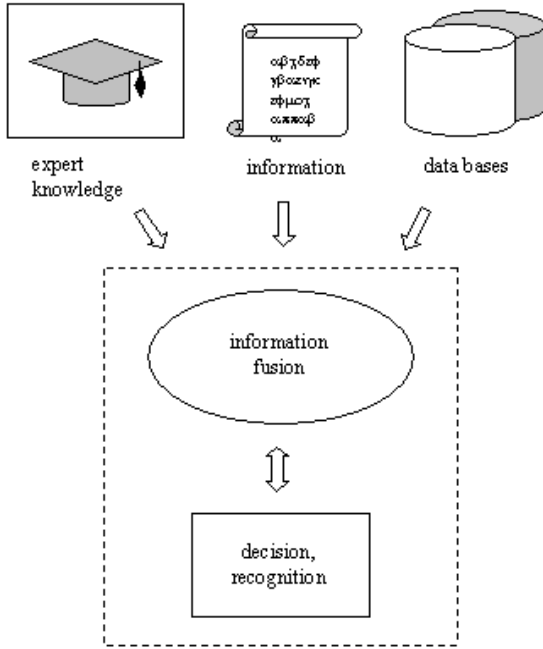


Fig. 1. Idea of information fusion

in the space of considered contours representing certain region in the image. The search is performed during evolution process (optimization) in which the given objective function, called energy, evaluates the quality of contour. Energy function is usually composed of two components, namely *internal* and *external* which express demands to the shape of contour and features of the ground, respectively.

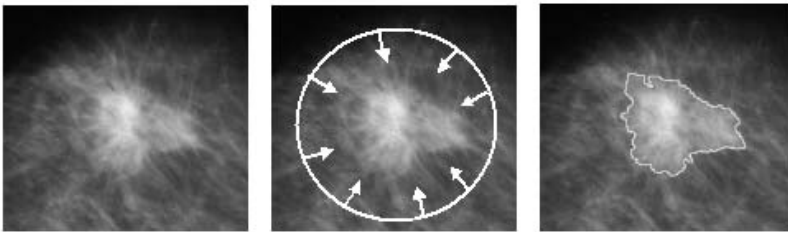


Fig. 2. Image segmentation using active contour

Moreover, contours are contextual classifiers of pixels (with a part of the pixels belonging to the interior and the other part - to the exterior of a given contour) and active contours are methods of optimal construction of classifiers [17, 18].



## 2 Adaptive Potential Active Hypercontours

Let us consider potential active hypercontours (PAH) - an implementation of active hypercontours introduced in [17]; see also [18, 19].

Let  $X$  - called feature space - be a metric space with metric  $\rho : X \times X \rightarrow \mathbb{R}$ . The potential hypercontour  $h$  is determined by set  $D^c$  of labeled control points and assigned to them potential functions  $P_i^c, i = 1, 2, \dots, N^c$  (Fig. 3).

Each point  $\mathbf{x}_i^c$  is a source of potential (similar to the electrical one), the value of which decreases as the distance from that point grows. There are many ways to define  $P_i^c$ , for example:

- exponential potential function

$$P_i^c = \Psi_i \exp(-\mu_i \rho^2(\mathbf{x}_i^c, \mathbf{x}))$$

- inverse potential function

$$P_i^c = \frac{\Psi_i}{1 + \mu_i \rho^2(\mathbf{x}_i^c, \mathbf{x})}$$

where real numbers  $\Psi_i$  and  $\mu_i$  are parameters characterizing the potential field of each control point. Note that the number of control points, their distribution and field parameters fully determine the shape of hypercontour.

Since each control point is labeled, i.e. one disposes of pairs  $(\mathbf{x}_i^c, l_i^c)$ , then  $h$  is in fact a classifier and it can be formally defined as follows:

$$C(\mathbf{x}) = \arg \max_{l \in L} \sum_{i=1}^N P_i^c(\mathbf{x}_i^c, \mathbf{x}) \delta(l_i^c, l)$$

where  $L$  is set of labels,  $l_i^c, l$  denote labels of  $\mathbf{x}_i^c$  and the examined one, respectively, while:

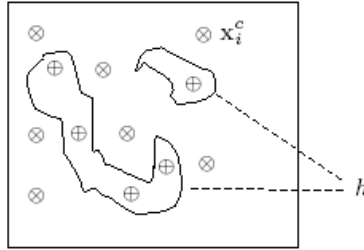
$$\delta(l_i^c, l) = \begin{cases} 0 & \text{for } l_i^c \neq l \\ 1 & \text{for } l_i^c = l \end{cases}$$

Potential active hypercontour (PAH) possesses the ability of *evolution* with change of the location of control points, and with modification of parameters of potential functions. The search of optimal hypercontour is performed by optimization of some performance index  $E$  called *energy* in the theory of active contours;  $E : H \rightarrow \mathbb{R} \cup \{0\}$  with  $H$  being the space of all available hypercontours. It has been proven in [17] that each hypercontour generates the corresponding classification function. This statement is true if the space  $X$  is metric. In  $E$  almost any type of information can be used assuming that we are able to implement this information in the computer oriented form. We can also decide if the classification is *supervised* or *unsupervised*. The present paper deals with the supervised one because of its more intuitive realization.

As mentioned above, there is certain freedom in defining of energy function  $E$ . However, we aim at incorporation of expert knowledge into  $E$  and there are many ways to do it. For example:

$$E(h) = \sum_{j=1}^J \chi_j [1 - \delta (l_j^{cor}, C(\mathbf{x}_j^{cor}))]$$

for each hypercontour  $h \in H$  where  $C$  is a corresponding to  $h$  classifier. Here,  $\{(\mathbf{x}_j^{cor}, l_j^{cor})\}_{j=1, \dots, J}$  denotes the set of correctly labeled objects (*reference concepts* or *reference points*),  $\chi_j$  is a real number introduced by expert and interpreted as significance of  $j$ -th reference point for correct classification, and  $\delta$  is the Kronecker symbol.



**Fig. 3.** Hypercontour  $h$  determined by control points  $\{\otimes, \oplus\}$ ; binary classification case

A similar concept of using expert opinion was proposed in the unsupervised classification problem [20] where expert estimated proximity between pairs of objects from the training set.

The search for optimal hypercontour may be conducted in many ways - e.g. by the use of simulated annealing or genetic algorithm; both perform global search and do not use gradient. In this work, we apply the former method.

Another interesting and powerful mechanism is *adaptation*. Discrimination ability of the given hypercontour is limited and it depends on the number of control points (assuming that other parameters are fixed). Flexibility of the potential active hypercontours (PAH) can be improved if we incorporate the change of the number of control points into the optimization procedure. The rate of misclassification in some area of space  $X$  can be the reason for introducing a few new control points.

### 3 Fusion

As shown in [17, 18] respectively, active hypercontours and active contours are contextual classifiers. Notice that the form of energy determines the classifier (assuming that the form of the potential function has been fixed).

When considering the particular active hypercontours presented in Section 2, we state that the contour is determined by the set of control points  $\mathbf{x}_i^c$  ( $i = 1, 2, \dots, N^c$ ). Looking for analogy in the neural networks theory, we can tell

that control points play the role of weights, while reference points  $\mathbf{x}_j^{cor}$  ( $j = 1, 2, \dots, J$ ) create the learning (or training) set of known correctly labeled objects ( $l_j^{cor}$  denotes the corresponding label).

Moreover, it is important to mention that expert knowledge is incorporated in the formula of energy. Consequently, we state that in the active hypercontour approach fusion is performed on three sources of information: expert knowledge (reflected in the formula of the energy), information reflected in  $\chi_j$ , data - reference points, cf. Fig. 1.

Notation:

- $E^k$  - formula of energy used for construction of the classifier  $C^k$ ;
- $\nu$  - number of iteration within the active hypercontour algorithm ( $\nu = 0, 1, 2, \dots$ );
- $E_\nu^k = E_\nu^k(h_\nu)$  - value of energy in the  $\nu$ -th iteration.

Taking certain form of energy function  $E^k$ , we aim at construction of the corresponding classifier  $C^k$  and thereby we decide about the actual form of the hypercontour which we are looking for. Due to its activity, the method generates a sequence of hypercontours. Each  $\nu$ -th proposed hypercontour  $h_\nu$  is evaluated by the energy function of the form  $k$ . If the quality of current hypercontour is not good enough ( $E_\nu^k$  is too large), then the hypercontour is modified in such a way that it should better fulfill the expectation closed in the form of the energy - in other words, the next possibly better hypercontour is created. We can perform the process with another form of energy, i.e. use another classifier  $l$ ,  $l \neq k$ .

### 3.1 Classifier Ensembles

Within this strategy, all available classifiers are working simultaneously on parallel computers, if available. Each  $\nu$ -th proposed hypercontour is evaluated by the energy function  $E^k$  of the form  $k$  and after each  $\nu$ -th iteration we dispose of  $K$  hypercontours and  $K$  values  $E_\nu^k$ . Two criteria for finishing classification can be applied:

- number of iterations which is common for all  $K$  processes;
- separate minimization of each energy function according to its specific criterion.

Once the computations have been finished the best result can be chosen or ranking of results can be created. Additionally, any known technique of classifiers' fusion, such as voting can be applied.

### 3.2 Quasi-competitive Classification

The actual  $\nu$ -th hypercontour is evaluated by all available energy functions  $E^k$  ( $k = 1, 2, \dots, K$ ). However, we decide for one particular classifier  $p$  determined by the hypercontour which is driven by the energy function  $E^p$ . We can start ( $\nu = 0$ ) with randomly chosen  $p$  or chose this form of energy for which  $E_0^k$  is minimal. After some number of iterations, say  $M$ , we may observe that the hypercontour

fulfills requirements of another, 'strange' energy function  $E^o$  ( $o \neq p$ ) better than of its own energy  $E^p$  currently used for making decisions. This is possible because modifications of the hypercontour depend on both internal and external energy. If the ground (external energy) does not allow some modifications forced by the internal energy, then the actual value  $E_v^p$  may not be the smallest one. Since the currently chosen classifier is not good enough, it is reasonable to evaluate another one, here determined by  $E^o$ , during next  $M$  iterations.

The quasi-competitive recognition seems to be close to human perception. If one concept seems to be better than another, we try to confirm the new hypothesis.

## 4 Concluding Remarks

Adaptive potential active hypercontour method can be interpreted in terms of information fusion. The reason is that the method uses three forms of information sources: expert knowledge, information and data. Moreover, by the organization of computations one can obtain classifier ensembles and any known technique, such as voting can be applied to increase the quality of final classification. The quasi-competitive classification have also been described. Obviously, other similar variants or combination of both organization of computations can also be proposed.

In case of image segmentation the main problems are construction of relevant energy function and incorporation of human knowledge about possible content of images [21]. In practice, the system must be restricted to well-defined and relatively small domain of shapes. The same problems must be addressed if active hypercontours are considered.

## References

1. Ho T.K., Hull J.J., Srihari S.N. (1994): *Decision Combination in Multiple Classifier Systems*. IEEE Trans. on Pattern Analysis and Machine Learning, vol. 41, pp.723-744.
2. Bauer E., Kohavi R. (1999): *An Empirical Comparison of Voting Classification Algorithm: Bagging, Boosting and Variants*. Machine Learning, vol.36, pp.105-142.
3. Ji C., Ma S. (1997): *Combination of Weak Classifiers*. IEEE Trans. on Pattern Analysis and Machine Learning, vol.8, pp.32-42.
4. Kittler J., Hatef M., Duin R.P.W., Matas J. (1998): *On Combining Classifiers*. IEEE Trans. on Pattern Analysis and Machine Learning, vol.20, pp.226-239.
5. Kuncheva L.I., Jain L.C. (2000): *Designing Classifier fusion Systems by Genetic Algorithms*. IEEE Trans. on Evolutionary Computation, vol.4, pp.327-336.
6. Kuncheva L.I., Jain L.C. (2002): *Switching between Selection and Fusion in Combining Classifiers: An Experiment*. IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics, vol.32, pp.156-150.
7. Kuncheva L.I., Whitaker C.J., Shipp C.A., Duin R.P.W. (2003): *Limits on the MajorityVote Accuracy in Classifier Fusion*. Pattern Analysis and Applications, vol.6, pp.22-31.

8. Kuncheva L.I. (2004): *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, New Jersey.
9. Kurzyski M., Woniak M. (2000): *Rule-based Algorithms with Learning for Sequential Recognition Problem*. Proc. of the Third Int. Conference on Information Fusion - Fusion'2000, Paris, France, TuC5: 24-29.
10. Kurzyski M. (2004): *Consistency Conditions of Expert Rule Set in the Probabilistic Pattern Recognition*. Lecture Notes in Computer Science, LNCS-3314, pp.831-836.
11. Kurzyski M. (2005): *Combining Rule-based and Sample-based Classifiers - Probabilistic Approach*. Lecture Notes in Computer Science, LNCS-3704, pp.298-307.
12. Lam L. (2000): *Classifier Combinations: Implementations and Theoretical Issue*. Lecture Notes in Computer Science, LNCS-1857, pp.77-87.
13. Woniak M. (2006): *Metody fuzji informacji dla komputerowych systemw rozpoznawania (Methods for information fusion for computer recognition systems)*; in Polish. Oficyna Wydawnicza Politechniki Wrocawskiej, Wrocaw.
14. Grzeszczuk R., Levin D. (1997): *Brownian Strings: Segmenting Images with Stochastically Deformable Models*. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19, no.10, 100-1013.
15. Kass M., Witkin W., Terzopoulos S. (1988): *Snakes: Active Contour Models*. Int. Journal of Computer Vision, vol. 1, no 4, 321-333.
16. Caselles V. Kimmel R., Sapiro G. (2000): *Geodesic Active Contours*. Int. Journal of Computer Vision, 22(1), 61-79.
17. Tomczyk A. (2005): *Active Hypercontours and Contextual Classification*. Proceedings of the 5th International Conference on Intelligent Systems Design and Applications - ISDA'2005. Wroclaw, Poland, IEEE Computer Society Press, pp.256-261.
18. Tomczyk A., Szczepaniak P.S. (2005): *On the Relationship between Active Contours and Contextual Classification*. In: M.Kurzyski, et al. (Eds.): *Computer Recognition Systems*. Proceedings of the 4th Int. Conference on Computer Recognition Systems - CORES'05. Springer, Berlin, Heidelberg, New York, pp. 303-310.
19. Tomczyk A., Szczepaniak P.S. (2006): *Adaptive Potential Active Hypercontours*. Proceedings of the 8th International Conference on Artificial Intelligence and Soft Computing - ICAISC'2006. Zakopane, Poland. Springer-Verlag, Berlin, Heidelberg; pp. 692-701.
20. Pedrycz W., Loia V. (2004): *P-FCM: A proximity-based fuzzy clustering*. Fuzzy Sets and Systems, vol. 128, pp.21-41.
21. Tomczyk A. (2007): *Image Segmentation using Adaptive Potential Active Contours*. In: *Computer Recognition Systems*. Proceedings of the 5th Int. Conference on Computer Recognition Systems - CORES'07 (this volume).

---

# Effective Ad Recognition Using Schur-type Signal Parametrization

Pawel Biernacki

Signal Theory Section, Technical University of Wroclaw, Wyb. Wyspianskiego 27,  
50-340 Wroclaw, Poland  
pawel.biernacki@pwr.wroc.pl

**Summary.** In this article an efficient audio ad recognition procedure is presented. Proposed algorithm uses the multidimensional orthogonal representation of the audio signal to parametrize the ad. Simulation results are presented for different recording data conditions.

## 1 Introduction

One of the main function of the media research companies is the radio and television ads monitoring. To investigate what and when is broadcast the efficient method of ad recognition is needed. In this article a new method is proposed which can be used for for the poor quality audio signals.

## 2 Multidimensional Audio Signal Parametrization

Given a vector  $|y \rangle_T$  of samples  $\{y_0, \dots, y_T\}$  of a time-series (a voice+music signal), observed on a finite time interval  $\{0, \dots, T\}$ , the signal parametrization problem can be stated as follows (see Figure 1). The estimate of the desired signal

$$|\hat{x}_N^{\{M\}} \rangle_T \triangleq \mathbf{P}(S_T)|x \rangle_T \quad (1)$$

is the orthogonal projection of the element  $|x \rangle_T$  on the space  $S_T$  spanned by the following set of the linear and nonlinear observations

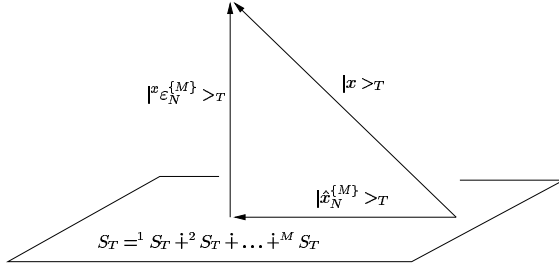
$$|Y \rangle_T = [|^1Y \rangle_T \ |^2Y \rangle_T \ \dots \ |^MY \rangle_T] \quad (2)$$

where

$$\begin{aligned} |^mY \rangle_T &= [|y_{i_1} \dots y_{i_m} \rangle_T; \ i_1 = 0, \dots, N, \\ &\quad i_2 = i_1, \dots, N, \dots, i_m = i_{m-1}, \dots, N] \end{aligned} \quad (3)$$

for  $m = 1, \dots, M$ . The orthogonal projection operator on  $|Y \rangle_T$  is defined as

$$\mathbf{P}(S_T) \triangleq |Y \rangle_T \langle Y|Y \rangle_T^{-1} \langle Y|_T \quad (4)$$



**Fig. 1.** The estimate  $|\hat{x} >_T$  of the desired signal

If an ON (generalized; i.e., multidimensional) basis of the space  $S_T$  is known, the projection operator on  $S_T$  can be decomposed as

$$\mathbf{P}(S_T) = \sum_{j_1=0}^N \mathbf{P}(|r_0^{j_1} >_T) + \dots + \sum_{j_1=0}^N \dots \sum_{j_M=j_{M-1}}^N \mathbf{P}(|r_0^{j_1, \dots, j_M} >_T) \tag{5}$$

where  $\mathbf{P}(|r_0^{j_1, \dots, j_m} >_T)$  stands for the orthogonal projection operator on the one-dimensional subspace spanned by the element  $r_0^{j_1, \dots, j_m}$ ,  $m = 1, \dots, M$  of an ON basis of the space  $S_T$ . Since

$$\mathbf{P}(|r_0^{j_1, \dots, j_w} >_T) = |r_0^{j_1, \dots, j_w} >_T \langle r_0^{j_1, \dots, j_w} |_T \tag{6}$$

the orthogonal expansion of the estimate of the desired signal can be written as

$$|\hat{x}_N^{\{M\}} >_T = \mathbf{P}(S_T)|x >_T = \sum_{j_1=0}^N |r_0^{j_1} >_T \langle r_0^{j_1} |x >_T + \dots + \sum_{j_1=0}^N \dots \sum_{j_M=j_{M-1}}^N |r_0^{j_1, \dots, j_M} >_T \langle r_0^{j_1, \dots, j_M} |x >_T \tag{7}$$

The estimation error associated with the element  $|\hat{x}_N^{\{M\}} >_T$  is then

$$|x_{\epsilon_N}^{\{M\}} >_T \triangleq \mathbf{P}(S_T^\perp)|x >_T = |x >_T - |\hat{x}_N^{\{M\}} >_T \perp S_T \tag{8}$$

The estimate (1) will be called optimal (in the least-squares sense) if the norm

$$\| |x_{\epsilon_N}^{\{M\}} >_T \| = \langle x_{\epsilon_N}^{\{M\}} |x_{\epsilon_N}^{\{M\}} >_T \rangle^{\frac{1}{2}} \tag{9}$$

of the estimation error vector (8) is minimized for each  $T = 0, 1, 2, \dots$

The multidimensional signal parametrization problem can be solved by the derivation of a (generalized) ON basis of the estimation space  $S_T$  (i.e. calculation of the orthogonal representation (the generalized Fourier coefficients) of the vector  $|x \rangle_T$  in the orthogonal expansion (7)).

To derive the desired ON basis of the estimation space  $S_T$ , we employ (consult [1]) the following

**Theorem 2.1** *The partial orthogonalization step results from the recurrence relations*

$$|e_{i_1, \dots, i_q}^{j_1, \dots, j_w} \rangle_T = [|e_{i_1, \dots, i_q}^{j_1, \dots, j_w-1} \rangle_T + |r_{i_1, \dots, i_q+1}^{j_1, \dots, j_w} \rangle_T \rho_{i_1, \dots, i_q; T}^{j_1, \dots, j_w}] (1 - (\rho_{i_1, \dots, i_q; T}^{j_1, \dots, j_w})^2)^{-\frac{1}{2}} \tag{10}$$

$$|r_{i_1, \dots, i_q}^{j_1, \dots, j_w} \rangle_T = [|e_{i_1, \dots, i_q}^{j_1, \dots, j_w-1} \rangle_T \rho_{i_1, \dots, i_q; T}^{j_1, \dots, j_w} + |r_{i_1+1, \dots, i_q+1}^{j_1, \dots, j_w} \rangle_T] (1 - (\rho_{i_1, \dots, i_q; T}^{j_1, \dots, j_w})^2)^{-\frac{1}{2}} \tag{11}$$

where

$$\rho_{i_1, \dots, i_q; T}^{j_1, \dots, j_w} = - \langle e_{i_1, \dots, i_q}^{j_1, \dots, j_w-1} | r_{i_1, \dots, i_q+1}^{j_1, \dots, j_w} \rangle_T \tag{12}$$

Proof can be found in [1].

The above relations make it possible to construct an orthogonal parametrization (decorrelation) filter, operating directly on the signal samples. The coefficients

$$\rho_{0; T}^{j_1, \dots, j_w} \tag{13}$$

can be interpreted as generalized Fourier coefficients which represent parametrized signal in the multidimensional space.

The above recurrence relations (11) actually solve the problem of the real-time derivation of the (generalized) ON basis of the estimation space. The diagram of the signal parametrization filter is presented in Figure 2.

The Schur coefficients (13) can be used for audio signal parametrization. Entire audio commercial parametrization procedure can be done in the tree steps:

1. dividing audio ad signal into the one second long pieces

$$y_{T_k} = \{y_T(n); n = k * f_P, k * F_P + 1, \dots, k * 2 * f_P - 1\} \tag{14}$$

where  $k$  means  $k$ -th signal section and  $k = 0, 2, \dots, K - 1$ .

2. computing Schur coefficients (13) vector for each piece
3. construct ID matrix for the audio commercial (using Schur coefficients vectors as its rows)

$$YM = \begin{pmatrix} \rho_{0; T_0}^0 & \dots & \rho_{0; T_1}^{j_1, \dots, j_w} & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \rho_{0; T_{K-1}} & \dots & \rho_{0; T_K}^{j_1, \dots, j_w} & \dots \end{pmatrix} \tag{15}$$



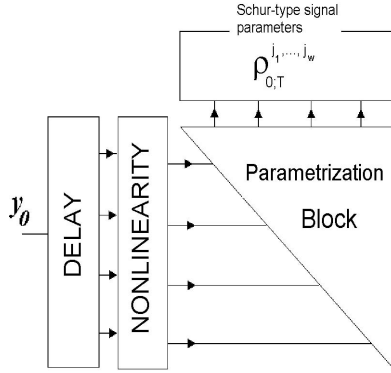


Fig. 2. Nonlinear orthogonal signal parametrization filter

To identify the one audio commercial the distances between pattern commercial matrix vectors and identifying commercial matrix vectors is computing by

$$d_i(YM_1, YM_2) = \sum_l |YM_1(i, l) - YM_2(i, l)| \tag{16}$$

for  $i = 0, 1, \dots, K - 1$  and  $l$  changes from 1 to matrix row length. If this distance for one or more  $i$  is higher then establish value the recognition is negative.

### 3 Results

The presented algorithm of the audio commercial recognition was implemented in C language and tested in PC environment. The following signals were tested:

- radio FM station commercials
- audio track from TV commercials

All ads were 8-bit signals sampled with frequency 11.2kHz. The following elements affect on recognition process:

- signals different levels
- different signal recording systems
- because before recognition process the commercial spot is divided into single ads, one ad can have a fragment of other ad at its beginning and/or end, or one ad can be cut at its beginning and/or end.

Table (3) shows influence of the filter parameters (filter order  $N$  and nonlinearity degree  $M$ ) on recognition effectiveness. Increasing values of the filter parameters above some values does not correct the recognition effectiveness, but can make it worse. It seems that filter order about  $N = 20$  and degree nonlinearity  $M = 2$  is the best solution.

**Table 1.** Influence of the filter parameters on the recognition effectiveness

Filter parameters	Recognition [%]
N=10, M=1	84
N=10, M=2	86
N=20, M=1	97
N=20, M=2	98
N=20, M=3	92
N=30, M=2	95

N - filter order, M - degree of nonlinearity

**Table 2.** Influence of the audio signal level on the recognition effectiveness. (N=20, M=2).

Filter parameters	Recognition [%]
signal 8-bits	98
signal 7-bits	91
signal 6-bits	51

Table (3) shows influence of the audio signal level on the recognition effectiveness. High signal level is a necessary condition for good recognition effectiveness.

During simulation one could find that the type of the recording system (type of the A/C card) does not influence on the recognition effectiveness for the proposed algorithm.

To make possible recognition for cutting ads or ads with the other ads fragments at their beginnings and/or their endings the cross-correlation function is used to establish the shift between compared ads. The shift for which there is the cross-correlation function maximum is used to position the compared ads. After ads positioning and adequate signal fragments parametrization the recognition effectiveness is about 98% ( $N = 20, M = 2$ ).

## 4 Conclusions

The presented results allow for the following conclusions:

- Presented algorithm is efficient in more than 98
- Presented solution is resistant on the recording systems differences
- The algorithm demand good signal level (more than 6-bits for sample)
- Cutting ads or ads with the other fragments do not make the recognition effectiveness worse
- Proper parametrization filter parameters selection is necessary for the high recognition effectiveness.

## References

1. P. Biernacki, J. Zarzycki, *Multidimensional Nonlinear Noise-Cancelling Filters of the Volterra-Wiener Class*, Proc. 2-Nd Int. Workshop on Multidimensional (nD) Systems (NDS-2000), ed. Inst. of Control and Comp. Eng. TU of Zielona Gora Press, Czocho Castle, pp. 255-261, 2000
2. P. Biernacki, J. Zarzycki, *Orthogonal Schur-Type Solution of the Nonlinear Noise-Cancelling Problem*, Proc. Int. Conf. On Signals and Electronic Systems (IC-SES'2000), Ustron, pp. 337-342, 2000
3. D.T.L. Lee, M. Morf, B. Friedlander, *Recursive Least-Squares Ladder Estimation Algorithms*, IEEE Trans. on CAS, Vol. 28, pp. 467-481, 1981
4. S. Schetzen, *The Volterra & Wiener Theories of nonlinear systems*, John Wiley & Sons, New York 1980

---

# Feature Extraction of Gray-Scale Handwritten Characters Using Gabor Filters and Zernike Moments

Ryszard S. Choraś

Faculty of Telecommunications University of Technology & Life Sciences  
85-796 Bydgoszcz, S. Kaliskiego 7, Poland  
choras@utp.edu.pl

**Summary.** In this paper a method for recognition of handwritten characters based on Gabor filters and Zernike moments is discussed. The technique can be summarized as follows: in the preprocessing stage the input image is first normalized. Next the feature extraction is performed. Thus, the image is represented by the feature vector. Mean and standard deviation of the magnitude of the Gabor transform coefficients and the Zernike moments are used as features. Classification is then carried out on the basis of the extracted features.

## 1 Introduction

Handwritten recognition have been a main research subject in pattern recognition field. Application of handwritten character recognition is broad. The recognition of handwritten characters, like other problems in pattern recognition, consists of two major problems: feature selection and pattern classification. Feature selection is problem-dependent and considered most central to the final result of a recognition system [9]. Since handwritten characters of the same character class can occur in great variety, it is desirable to generate a representation that is invariant. A feature-based recognition of objects which is independent of their position, size, orientation and other variations has been the goal of recent research.

In this paper, a new method of feature extraction for the handwritten character recognition problem from an early vision point of view is introduced. A set of spatial Gabor filters as receptive fields to operate on the handwritten character images of multiple resolution scales is applied. Different responses for different local spatial frequencies and directions are obtained. Features extracted by Gabor filters (we call them Gabor features) have been successfully applied to many pattern recognition applications such as character recognition. It seems that Gabor features are good candidates for handwritten character recognition because: 1) Gabor features are well motivated and mathematically well-defined, 2) they are easy to understand and fine-tune.

Next, we implemented Zernike moments as feature descriptors in the application of handwritten character recognition. Character image can be better described by a small set of its Zernike moments than any other type of moments such as geometric moments, rotational moments, and complex moments in terms of mean-square error.

The feature vector is constructed using Gabor features and Zernike moments.

## 2 Preprocessing

The pictorial information is represented as a function of two variables  $(i, j)$ . The image in its digital form is usually stored as an two-dimensional array. If  $N = \{0, 1, \dots, N - 1\}$  then  $D = N \times N$  is the set of resolutions cells and the digital image  $I$  is a function which assigns gray value to each and every resolution cells, i.e.  $I : N \times N \rightarrow G$ .

We perform normalizations on size before feature extraction. Because the Zernike moments introduce a set of complex polynomials which form a complete orthogonal set over the interior of a unit circle, i.e.,  $x^2 + y^2 \leq 1$ , the image  $(N \times N)$  pixels is transformed and normalized over unit circle; i.e.  $x^2 + y^2 \leq 1$ , which the transformed unit circle image is bounding the square image (Figure 1) [4], [7]. We have

$$\begin{aligned}
 r_{ij} &= \sqrt{x_i^2 + y_j^2}; \quad \theta = \tan^{-1}\left(\frac{y_j}{x_i}\right) \\
 x_i &= \frac{\sqrt{2}}{N-1}i + \frac{-1}{\sqrt{2}}; \quad y_j = \frac{\sqrt{2}}{N-1}j + \frac{-1}{\sqrt{2}}
 \end{aligned} \tag{1}$$

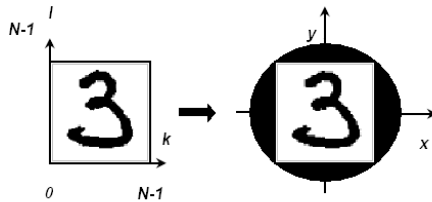


Fig. 1. Image transformation

## 3 Gabor Filter

The general functionality of the 2D Gabor filter family can be represented as a Gaussian function modulated by a complex sinusoidal signal [3]. Specially, a 2D Gabor filter  $g(x, y)$  can be formulated as

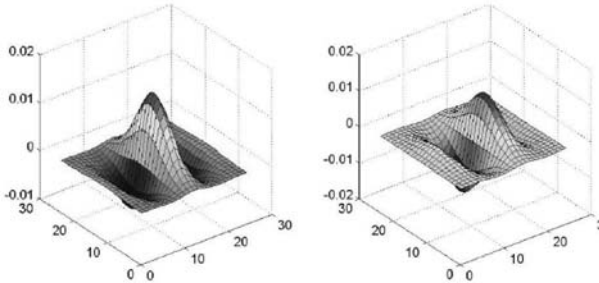
$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{\bar{x}^2}{\sigma_x^2} + \frac{\bar{y}^2}{\sigma_y^2}\right)\right] \exp[2\pi jW\bar{x}] \tag{2}$$

where

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}, \quad j = \sqrt{-1}$$

and

- $\sigma_x$  and  $\sigma_y$  are the scaling parameters of the filter and determine the effective size of the neighborhood of a pixel in which the weighted summation (convolution) is performed,
- $\theta$  ( $\theta \in [0, \pi]$ ) specifies the orientation of the Gabor filters,
- $W$  is the radial frequency of the sinusoid.



**Fig. 2.** Real (a) and imaginary (b) parts of Gabor wavelets

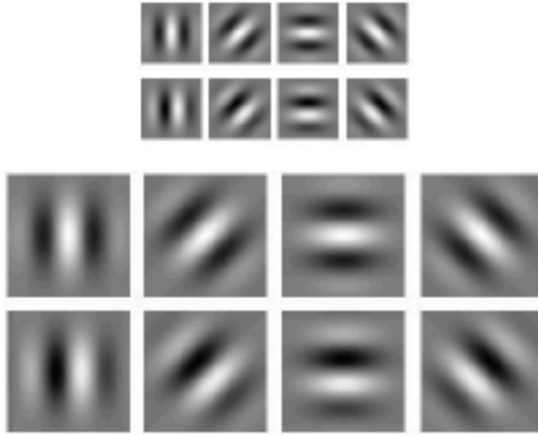
The Gabor filter has the following properties: tunable to specific orientations; spectrally localised, which allows differentiation of linear patterns occurring over a range of spatial scales or channels; spatially localised, which allows accurate identification of linear patterns within a specific orientation. It has been shown that the complex Gabor function achieves the lower bound of the uncertainty principle. The 2D Gabor filter has the form of a complex sinusoid multiplied by a 2D Gaussian filter. Gabor filters extract local information from the image, as opposed to global techniques such as the Fourier transform.

In our work we use a bank of filters built from these Gabor functions for handwritten character feature extraction. Before filtration, we normalize an image to remove the effects of sensor noise and gray level deformation.

Gabor filters worked as local bandpass filters and each filter is fully determined by choosing the four parameters  $\{\theta, W, \sigma_x, \sigma_y\}$ . Assuming that  $N$  filters are needed in an application,  $4N$  parameters need to be optimized. The orientation parameter  $\theta$  should satisfy  $\theta \in [0, \pi]$ .  $W$  is the radial frequency of the Gabor filter and is application dependent.  $\sigma_x$  and  $\sigma_y$  are the effective sizes of the Gaussian functions and are within the range  $[\sigma_{min}, \sigma_{max}]$ .

Fourier transform of the two-dimensional Gabor function (2) can be written as

$$H(u, v) = \exp \left\{ -\frac{1}{2} \left[ \frac{(u - W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right] \right\} \tag{3}$$



**Fig. 3.** Even-symmetric and odd-symmetric components of Gabor filters (two scales and four orientations)

To eliminate sensitivity of the filter response to absolute intensity values, the real (even) components of the 2-D Gabor filters are biased by adding a constant to make them zero mean. Each channel is formed by a pair of real Gabor filters. Let the output of each channel be given by

$$C_{ev}(x, y; W, \theta) = g(x, y) \cos(2\pi W, x') * I(x, y) \tag{4}$$

$$C_{odd}(x, y; W, \theta) = g(x, y) \sin(2\pi W, x') * I(x, y) \tag{5}$$

The channel output is computed as

$$C(x, y; W, \theta) = \sqrt{C_{ev}^2 + C_{odd}^2} \tag{6}$$

Filters are implemented in frequency domain for better computational efficiency. The mean value  $\mu(W, \theta)$  of a channel is computed by

$$\mu(W, \theta) = \frac{1}{K_1 K_2} \sum C(x, y; W, \theta) \tag{7}$$

where  $K_1 K_2$  is the area of  $C(x, y; W, \theta)$  and standard deviation  $S(W, \theta)$

$$S(W, \theta) = \frac{\sqrt{\sum (C(x, y; W, \theta) - \mu(W, \theta))^2}}{K_1 K_2} \tag{8}$$

This value depends on the filter center frequency  $W$  and orientation  $\theta$ . The mean values provide powerful features for texture classification. The features are rotation dependent since  $\mu(W, \theta_i) \neq \mu(W, \theta_j)$  for  $i \neq j$ .

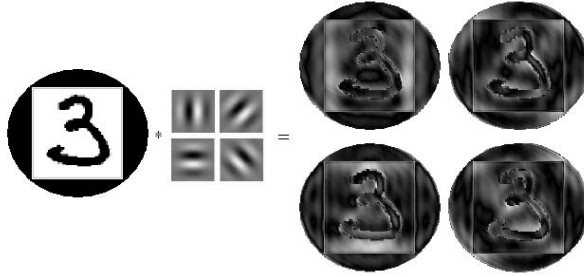


Fig. 4. Gabor filters convolution

### 4 Zernike Moments Descriptors

In this paper we had implemented Zernike moments (ZM) as feature descriptors in the application of handwritten characters recognition [1],[5]. Orthogonality property of the ZM enables redundancy reduction among their respective description and improve to computation efficiency.

The kernel of Zernike moments is a set of orthogonal Zernike polynomials defined over the polar coordinate space inside a unit circle. The two-dimensional Zernike moments of order  $p$  with repetition  $q$  of an image intensity function  $f(r, \theta)$  are defined as [4],[10]

$$Z_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_0^1 V_{pq}(r, \theta) f(r, \theta) r dr d\theta; \quad |r| \leq 1 \tag{9}$$

where Zernike polynomials  $V_{pq}(r, \theta)$  are defined as

$$V_{pq}(r, \theta) = R_{pq}(r) e^{-\hat{j}q\theta}; \quad \hat{j} = \sqrt{-1} \tag{10}$$

and the real-valued radial polynomials,  $R_{pq}(r)$  is defined as follows

$$R_{pq}(r) = \sum_{k=0}^{\frac{p-|q|}{2}} (-1)^k \frac{(p-k)!}{k! (\frac{p+|q|}{2} - k)! (\frac{p-|q|}{2} - k)!} r^{p-2k} \tag{11}$$

where  $0 \leq |q| \leq p$  and  $p - |q|$  is even.

If  $N$  is the number of pixels along each axis of the image, then the discrete approximation of Eq (9) is given by

$$Z_{pq} = \lambda(p, N) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R_{pq}(r_{ij}) e^{\hat{j}q\theta_{ij}} f(i, j) \quad 0 \leq r_{ij} \leq 1 \tag{12}$$

where  $\lambda(p, N)$  is normalizing constant

$$\lambda(p, N) = \frac{4(p+1)}{(N-1)^2 \pi} \tag{13}$$

$Z_{pq}$  is split into its real  $Z_{pq}^{real}$  and imaginary  $Z_{pq}^{imag}$  parts



$$\begin{aligned}
 Z_{pq}^r &= \frac{2(p+1)}{\pi} \int_0^{2\pi} \int_0^1 R_{pq}(r) \cos(q\theta) f(r, \theta) r dr d\theta \\
 Z_{pq}^i &= \frac{2(p+1)}{\pi} \int_0^{2\pi} \int_0^1 R_{pq}(r) \sin(q\theta) f(r, \theta) r dr d\theta
 \end{aligned}
 \tag{14}$$

where  $p \geq 0, q > 0$ .

Because  $Z_{mn}$  is complex, we use the Zernike moments modules  $|Z_{mn}|$  as the features of shape in the recognition of pattern.

The magnitude of Zernike moments has rotational invariance property.

## 5 Features Selection and Classification

To reduce dimensions of feature vectors we use LDA. LDA is a method to find a transformation that can maximize the between-class scatter matrix  $Scat_b$  and minimize the within-class scatter matrix  $Scat_w$  simultaneously

$$argD \left\{ \max \frac{D^T Scat_b D}{D^T Scat_w D} \right\}
 \tag{15}$$

where  $D$  is the new most discriminative space.

Feature vector  $FV$  in the original feature space is projected to this new space and have new feature  $NFV$

$$NFV = D^T FV
 \tag{16}$$

which is used for classification.

The feature vector which is build-up with the Gabor feature parameters and Zernike moments extracted from the input character image is projected onto the LDA space. A simple classification scheme as  $k$  nearest-neighbor ( $kNN$ ) is utilized effectively.

$NFV_i$  is ideal feature vector for each class handwritten characters in the training set. Unknown character is represented as  $NFV_u$ . The distance  $d_k$  of the unknown character from class  $k$  is given by

$$d_k = \left[ \sum_{j=1}^M (NFV_u - NFV_{kj})^2 \right]^{\frac{1}{2}}
 \tag{17}$$

where  $j = 1, \dots, M$  and  $M$  is the number of the features.

The character is assigned to the class  $v$  such that

$$d_v = \min(d_k).
 \tag{18}$$



## 6 Experimental Results

In our experiments, the number of filters is  $4 \times 4 = 16$  where  $s = 4$  is the number of scales (4, 8, 16 and 32 cycles/degree) and  $k = 4$  is the number of orientations ( $0^\circ, 45^\circ, 90^\circ$  and  $135^\circ$ ) used in the multi-resolution decomposition using Gabor functions. This gives a total 16 output images (4 for each frequency)

from which the features are extracted. These features are the mean and the standard deviation of each output image. Therefore 32 features per input image are calculated.



The mean and the standard deviation values of the Gabor filtered characters for various orientations are shown in the Table 1.

**Table 1.** Gabor features

Character	Feature	$\theta = 0^\circ$	$\theta = 45^\circ$	$\theta = 90^\circ$	$\theta = 135^\circ$
	Mean	186,35	197,51	188,13	195,90
	Std. dev.	46,668	39,648	46,271	40,16
	Mean	184,36	189,21	193,31	198,65
	Std. dev.	48,92	42,778	44,661	39,83

For some digital characters, the Zernike moments values for any order are presented in Table 2.

**Table 2.** Zernike moments

Character	$(pq)$	(8,2)	(8,4)	(8,6)	(10,6)	(10,8)
		83,3	16,2	1,4	10,4	1,8
		142,2	18,2	1,65	14,6	0,9

## 7 Conclusion

In this paper, we have studied Gabor features and Zernike moments and its properties and proposed its use for the recognition of handwritten characters. The recognition rate of using combinations of Gabor features and Zernike moments order 10 (34 moments) was found - 98.7%. For real time applications, we prefer only 32 Gabor features and 34 Zernike moments. When more feature sets were used, the recognition rate was found to be higher.

## References

1. Belkasim SO, Ahmadi M, Shridhar M (1996) Efficient algorithm for fast computation of zernike moments. In: IEEE 39th Midwest Symposium on Circuits and Systems, 3:1401–1404.
2. Flusser J, Suk T (1993) Pattern Recognition, 26:167–174.

3. Gabor D (1946) *Journal of the Institute of Electrical Engineers*, 93:429–457.
4. Khotanzad A, Hong YH (1990) *IEEE Trans. Pattern Anal. Machine Intell.*, 12(5):489–498.
5. Kim WY, Kim YS (2000) *Signal Processing: Image Communications*, 16:95–102.
6. Maitra S (1997) *Proc. IEEE*, 67:697–699.
7. Mukundan R, Ramakrishnan KR (1998) *Moment Functions in Image Analysis: Theory and Applications*. World Scientific Publication Co., Singapore.
8. Su YM, Wang JF (2003) *Pattern Recognition*, 36:635–647.
9. Suen CY, Nadal C, Mai TA, Legault R, Lam L (1992) *Proc. IEEE*, 80(7):1162–1189.
10. Teh CH, Chin RT (1988) *IEEE Trans. Pattern Anal. Machine Intell.*, (10):496–513.
11. Young IT, Vliet van LJ, Ginkel van M (2002) *IEEE Transactions on Signal Processing*, 50(11):2798–2805.

---

# Pattern Discovery Through Separable Data Projections

Leon Bobrowski<sup>1</sup>, Volodymir Mashtalir<sup>2</sup>, and Magdalena Topczewska<sup>1</sup>

<sup>1</sup> Faculty of Computer Science, Białystok Technical University, Poland

<sup>2</sup> Kharkov National University of Radio Electronics, Ukraine

**Summary.** Data projections or, more generally, data linear transformations, in some cases allow to enhance interesting regularities in data sets. We pay particular attention to linear transformations from multidimensional feature space on a line and on a plane. In such cases, transformed data sets can be visualized and the resulting patterns can be evaluated by an expert both analytically and subjectively in accordance with the expert's opinion. The projection pursuit provides well developed methods for designing interesting projections of data sets related to the normal model. Here we are considering separability criteria for designing projections.

## 1 Introduction

Discovering meaningful patterns in data sets or useful data models are the most important goals of exploratory data analysis. Variety of tools of multivariate statistical analysis [1] or pattern recognition [2] can be used for this purpose. Explored data is usually represented as feature vectors of the same dimensionality or as points in multidimensional feature space. If feature vectors have been assigned an a priori category then they can be grouped as learning sets linked to particular categories.

Data linear transformations, in some cases allow to enhance interesting patterns in data sets. Linear transformations based on the methods of principal components analysis [3] or discriminant analysis [4] can be used for this purpose. Projections of multidimensional feature vectors on a line or on a plane are special cases of linear transformations. *Projection pursuit* is well developed method of designing projections [5]. The "interesting" projections identified by this method takes into account deviations from the normal model.

Designing data projections can be also based on the *principle of separability* [6]. Data transformation should assure the dimensionality reduction while preserving the separability of the learning sets. Designing linear transformations in accordance with the principle of separability can be carried out through minimization of the convex and piecewise linear (*CPL*) criterion functions. The separable transformations give a chance for discovering valuable patterns in learning sets. These topics are analyzed in the presented paper.

## 2 Linearly Separable Sets

Let us assume that descriptions of objects stored in a given database are represented as the so-called feature vectors  $\mathbf{x} = [x_1, \dots, x_n]^T$ , or as points in the  $n$ -dimensional feature space  $\mathbf{X}[n]$  ( $\mathbf{x} \in \mathbf{X}[n]$ ). The components  $x_i$  of the vectors  $\mathbf{x}$  called *features* are numerical results of a variety of examinations or measurements of a given object. We can assume that the feature vectors  $\mathbf{x}$  can be of the mixed, qualitative-quantitative type. It means that each feature  $x_i$  ( $i = 1, \dots, n$ ) can take a binary value ( $x_i \in \{0, 1\}$ ) or a real value ( $x_i \in R$ ).

We assume that a given database contains descriptions  $\mathbf{x}_j(k)$  of  $m$  objects ( $j = 1, \dots, m$ ) labelled in accordance with one of the category (*class*)  $\omega_k$  ( $k = 1, \dots, K$ ). For example, a clinical database contains the descriptions of  $m$  patients  $\mathbf{x}_j(k)$  labelled in accordance with their clinical diagnosis  $\omega_k$ . As an another example - the categories  $\omega_k$  could stand for particular letters "a", "b", "c", ... In this case, the feature vectors  $\mathbf{x}_j(k)$  of particular category  $\omega_k$ , could represent different styles of writing or printing of the letter of given category. The learning set  $C_k$  contains  $m_k$  labelled feature vectors  $\mathbf{x}_j(k)$  related to the  $k$ -th class or category  $\omega_k$ :

$$C_k = \{\mathbf{x}_j(k)\} \quad (j \in I_k) \tag{1}$$

where  $I_k$  is the set of indices  $j$  of the feature vectors  $\mathbf{x}_j(k)$  belonging to the class  $\omega_k$ .

We are considering the separation of the sets  $C_k$  (1) by the hyperplanes  $H(\mathbf{w}_k, \theta_k)$ :

$$H(\mathbf{w}_k, \theta_k) = \{\mathbf{x} : (\mathbf{w}_k)^T \mathbf{x} = \theta_k\} \tag{2}$$

where  $\mathbf{w}_k = [w_{k1}, \dots, w_{kn}]^T \in R^n$  is the weight vector,  $\theta_k \in R^1$  is the threshold.

**Definition 1.** *The feature vector  $\mathbf{x}$  is situated on the positive side of the hyperplane  $H(\mathbf{w}_l, \theta_l)$  if and only if  $(\mathbf{w}_l)^T \mathbf{x} > \theta_l$ . Similarly, the vector  $\mathbf{x}$  is situated on the negative side of  $H(\mathbf{w}_l, \theta_l)$  if and only if  $(\mathbf{w}_l)^T \mathbf{x} < \theta_l$ .*

**Definition 2.** *The learning sets (1) are linearly separable if each of the sets  $C_k$  can be fully separated from the sum of the remaining sets  $C_{k'}$  by some hyperplane  $H(\mathbf{w}_k, \theta_k)$  (2):*

$$\begin{aligned}
 (\forall k \in \{1, \dots, K\}) \quad & (\exists \mathbf{w}_k, \theta_k) \quad (\forall \mathbf{x}_j(k) \in C_k) \quad (\mathbf{w}_k)^T \mathbf{x}_j(k) > \theta_k \\
 & \text{and } (\forall \mathbf{x}_{j'}(k') \in C_{k'}) \quad (\mathbf{w}_k)^T \mathbf{x}_{j'}(k') < \theta_k
 \end{aligned} \tag{3}$$

In accordance with the relation (3), all the vectors  $\mathbf{x}_j(k)$  belonging to the learning set  $C_k$  are situated on the positive side of the hyperplane  $H(\mathbf{w}_k, \theta_k)$  (2) and all the feature vectors  $\mathbf{x}_{j'}(k')$  from the remaining sets  $C_{k'}$  are situated on the negative side of this hyperplane.

## 3 Linear Transformations of Feature Vectors

We will take into consideration the projection of  $n$ -dimensional feature vectors  $\mathbf{x}_j(k)$  (1) on *lines*  $\mathbf{l}(\mathbf{w}_1)$ , and *planes*  $\mathbf{P}(\mathbf{w}_1, \mathbf{w}_2)$  defined by the  $n$ -dimensional

parameters vectors  $\mathbf{w}_i$  ( $\mathbf{w}_i \in R^n$ ) and passing through the point  $\mathbf{0}$  (the beginning of the coordinate system) in the  $n$ -dimensional feature space  $\mathbf{X}[n]$  ( $\mathbf{x} \in \mathbf{X}[n]$ ):

$$\mathbf{l}(\mathbf{w}_1) = \{\mathbf{x} : \mathbf{x} = t_1 \mathbf{w}_1\} \quad (4)$$

and

$$\mathbf{P}(\mathbf{w}_1, \mathbf{w}_2) = \{\mathbf{x} : \mathbf{x} = t_1 \mathbf{w}_1 + t_2 \mathbf{w}_2\} \quad (5)$$

where  $\mathbf{w}_i = [w_{i1}, \dots, w_{in}]^T \in R^n$ ,  $t_i \in R^1$ , ( $i = 1, 2$ ), and the parameters vectors  $\mathbf{w}_1$  and  $\mathbf{w}_2$  are not parallel.

The transformation of the  $n$ -dimensional feature vectors  $\mathbf{x}_j(k)$  (1) on the points  $\mathbf{y}'_j(k)$  ( $\mathbf{y}'_j(k) \in R^1$ ) of the line  $\mathbf{l}(\mathbf{w}_1)$  (4) defined by the parameter vector  $\mathbf{w}_1$  can be given as:

$$\mathbf{y}'_j(k) = \mathbf{w}_1^T \mathbf{x}_j(k) \quad (6)$$

Similarly, the feature vectors  $\mathbf{x}_j(k)$  (1) transformed on the plane  $\mathbf{P}(\mathbf{w}_1, \mathbf{w}_2)$  (5) can be represented as two-dimensional vectors  $\mathbf{y}'_j(k)$  ( $\mathbf{y}'_j(k) \in R^2$ ) on the *visualizing plane*:

$$\mathbf{y}'_j(k) = [y'_{1j}(k), y'_{2j}(k)]^T = [\mathbf{w}_1^T \mathbf{x}_j(k), \mathbf{w}_2^T \mathbf{x}_j(k)]^T \quad (7)$$

The representations (6) and (7) give the dimensionality reduction from  $n$  to one or two, adequately.

## 4 Two Models of Learning Sets

The below models of the learning sets  $C_k$  (1) have been presented here to illustrate the properties of projection transformations (6) and (7).

**Model 1:** Let us assume that there exists such an unknown parameter vector  $\mathbf{a} = [a_1, \dots, a_n]^T$  and unknown scalars  $\tau_k$  ( $k = 1, \dots, K$ ) that each feature vector  $\mathbf{x}_j(k)$  from the learning sets  $C_k$  (1) fulfills the below *ranked* conditions:

$$(\exists \mathbf{a} \in R^n)(\forall k \in \{1, \dots, K\})(\exists \tau_k \in R^1)(\forall \mathbf{x}_j(k) \in C_k) \quad (8)$$

$$\tau_k < \mathbf{a}^T \mathbf{x}_j(k) < \tau_{k+1}$$

where  $\tau_{k+1}$  can be chosen as  $\tau_{k+1} = \infty$ .

In accordance with the inequalities (8), all the feature vectors  $\mathbf{x}_j(k)$  from one learning set  $C_k$  (1) are contained in the layer (*slice*)  $L(\mathbf{a}, \tau_k, \tau_{k+1})$  in the feature space  $\mathbf{X}[n]$ , where

$$L(\mathbf{a}, \tau_k, \tau_{k+1}) = \{\mathbf{x} : \tau_k < \mathbf{a}^T \mathbf{x} < \tau_{k+1}\} \quad (9)$$

Learning sets  $C_k$  (1) *ranked* by the inequalities (8) can represent dynamical system or event when particular layers represent a temporal development. The parameter vector  $\mathbf{a}$  characterizes a *trend* in a development of a dynamical event. The inequalities (8) can describe a generation of the feature vectors  $\mathbf{x}_j(k)$  from successive "slices" in the feature space  $\mathbf{X}[n]$ . For example, the development

of the cancer can be characterized by a temporal sequence of images with the inequalities (8). The Cox model used in the survival analysis can also lead to such relations (8) [7].

*Remark 1:* The learning sets  $C_k$  (1) that fulfill the inequalities (8), are not linearly separable (3) if their number  $K$  is greater than two ( $K > 2$ ), but an arbitrary pair  $\{C_k, C_l, k \neq l\}$  of these sets is linearly separable (linear separability in pairs of sets).

The feature vectors  $\mathbf{x}_j(k)$  (1) can be projected on the line  $\mathbf{l}(\mathbf{a}_1)$  (4) in accordance with the equation (6). As a result, the learning sets  $C_k$  (1) are transformed into the sets  $C'_k$ :

$$C'_k = \{\mathbf{y}'_j(k)\} \quad (j \in I_k) \tag{10}$$

It can be seen that the transformed sets  $C'_k$  are linearly separable (3) in pairs  $\{C_k, C_l, k \neq l\}$ . In other words, the reduction of the dimensionality equal to  $n$  (potentially large) to dimensionality equal to one allows in this case to fully preserve the separability of the learning sets  $C_k$  (1).

**Model 2:** Let us assume the projection plane  $\mathbf{P}(\mathbf{a}_1, \mathbf{a}_2)$  (5) is defined by such unknown parameter vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  that each of these vectors  $\mathbf{a}_i$  ( $i \in \{1, 2\}$ ) assures the linear separability (3) of different sets  $G_i^+$  and  $G_i^-$  of the feature vectors  $\mathbf{x}_j(k)$  (1):

$$\begin{aligned} (\forall i \in \{1, 2\})(\exists \mathbf{a}_i)(\exists \theta_i)(\forall \mathbf{x}_j(k) \in G_i^+) \quad & (\mathbf{a}_i)^T \mathbf{x}_j(k) > \theta_i \\ \text{and } (\forall \mathbf{x}_j(k) \in G_i^-) \quad & (\mathbf{a}_i)^T \mathbf{x}_j(k) < \theta_i \end{aligned} \tag{11}$$

where

$$G_1^+ \neq G_2^+ \quad \text{and} \quad G_1^- \neq G_2^- . \tag{12}$$

**Lemma 1.** *If the projection plane  $\mathbf{P}(\mathbf{a}_1, \mathbf{a}_2)$  (5) is defined by such parameter vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  that each of the vectors  $\mathbf{a}_i$  ( $i = 1, 2$ ) assures the linear separability (11) of different sets  $G_i^+$  and  $G_i^-$  (12) of the feature vectors  $\mathbf{x}_j(k)$  (1), then the projected vectors  $\mathbf{x}_j(k)$  can be represented as points  $\mathbf{y}'_j(k)$  (7) on the visualizing plane with the below structure:*

$$\begin{aligned} \text{the upper - left quarter} & - G_1^- \cap G_2^+, \\ \text{the upper - right quarter} & - G_1^+ \cap G_2^+, \\ \text{the lower - left quarter} & - G_1^- \cap G_2^-, \\ \text{the lower - right quarter} & - G_1^+ \cap G_2^- . \end{aligned} \tag{13}$$

The above structure means among others that all the points  $\mathbf{y}'_j(k)$  (7) from the set  $G_1^- \cap G_2^+$  are located in the upper-left quarter of the visualizing plane.

The sketched models show a possibility of a significant reduction of dimensionality equal to  $n$  (possibly high) to dimensionality equal to 1 or 2 while preserving particular separability of the learning sets  $C_k$  (1). The problem is how to induce an unknown parameter vector  $\mathbf{a}$  (8) of the *Model 1* or the pair of the vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  (11) of the *Model 2* from the learning sets  $C_k$  (1). The minimization of the *CPL* criterion functions can be used for this purpose.

## 5 Convex and Piecewise Linear (CPL) Criterion Functions

Let us define the positive  $G^+$  and the negative  $G^-$  sets of feature vectors  $\mathbf{x}_j$  (1).

$$G^+ = \{\mathbf{x}_j\} \text{ where } j \in J^+ \text{ and } G^- = \{\mathbf{x}_j\} \text{ where } j \in J^- \quad (14)$$

where  $J^+$  is the positive set of indices  $j$ , and  $J^-$  is the negative set of indices  $j$  ( $J^+ \cap J^- = \emptyset$ ).

The *augmented vectors*  $\mathbf{z}_j$  are introduced on the basis of sets  $G^+$  and  $G^-$  [2]:

$$(\forall \mathbf{x}_j \in G^+) \mathbf{z}_j = [1, (\mathbf{x}_j)^T]^T \text{ and } (\forall \mathbf{x}_j \in G^-) \mathbf{z}_j = [-1, (-\mathbf{x}_j)^T]^T \quad (15)$$

**Lemma 2.** *The sets  $G^+$  and  $G^-$  (14) are linearly separable (3) if and only if there exists such parameter vector  $\mathbf{v}'$  that all the scalar products  $(\mathbf{v}')^T \mathbf{z}_j$  (15) are not less than one:*

$$(\exists \mathbf{v}' \in R^{n+1}) (\forall \mathbf{x}_j \in G^+ \cup G^-) (\mathbf{v}')^T \mathbf{z}_j \geq 1 \quad (16)$$

The below CPL penalty functions  $\varphi_j(\mathbf{v})$  are used in order to force the conditions (16):

$$\begin{aligned} (\forall \mathbf{x}_j \in G^+ \cup G^-) \varphi_j(\mathbf{v}) &= 1 - (\mathbf{v})^T \mathbf{z}_j \text{ if } (\mathbf{v})^T \mathbf{z}_j < 1 \text{ or} \\ \varphi_j(\mathbf{v}) &= 0 \text{ if } (\mathbf{v})^T \mathbf{z}_j \geq 1 \end{aligned} \quad (17)$$

The convex and piecewise linear (CPL) criterion function  $\Phi(\mathbf{v})$  is defined on the vectors  $\mathbf{z}_j$  (15) as the weighted sum of the penalty functions  $\varphi_j(\mathbf{v})$  [2, 6, 8]:

$$\Phi(\mathbf{v}) = \sum_{j \in J} \alpha_j \varphi_j(\mathbf{v}) \quad (18)$$

where  $\alpha_j$  ( $\alpha_j > 0$ ) are positive parameters (*prices*), and  $J = J^+ \cup J^-$  is the set of indices  $j$  (14).

Minimization of the function  $\Phi(\mathbf{v})$  defined on the vectors  $\mathbf{z}_j$  (15) allows to find such parameter vector  $\mathbf{v}^* = [-\theta^*, (\mathbf{w}^*)^T]^T$  which separates (11) two sets  $G^+$  and  $G^-$  (14) in the best way

$$\Phi^* = \Phi(\mathbf{v}^*) = \min_{\mathbf{v}} \Phi(\mathbf{v}) \geq 0 \quad (19)$$

The basis exchange algorithms which are similar to the linear programming, allow to find the minimum of the criterion function  $\Phi(\mathbf{v})$  efficiently even in cases of large, multidimensional data sets  $G^+$  and  $G^-$  (14) [9].

It has been proved that the minimal value  $\Phi^*$  ( $0 \leq \Phi^* \leq 1$ ) of the criterion function  $\Phi(\mathbf{v})$  (19) is equal to zero ( $\Phi^* = 0$ ) if and only if the positive  $G^+$  and the negative  $G^-$  sets (19) are linearly separable (3) [2].



## 6 Designing Separable Projections

Let us first consider the projections (6) on the line  $\mathbf{l}(\mathbf{w}_1)$  (4) defined by the parameter vector  $\mathbf{w}_1$ . The vector  $\mathbf{w}_1$  should be chosen in a way that will allow to preserve the separability of the sets  $C_k$  (1) on the line  $\mathbf{l}(\mathbf{w}_1)$  (4). In the case of the learning sets  $C_k$  (1) consistent with the conditions (11) of the *Model 2*, the separability of the sets  $C_k$  (1) is preserved if the parameter vector  $\mathbf{w}_1$  is equal or near to the unknown, separable vector  $\mathbf{a}$  of the assumed model (11). It has been shown that the *CPL* criterion function  $\Phi(\mathbf{v})$  (18) defined on the differential vectors  $\mathbf{r}_{jk} = (\mathbf{x}_k - \mathbf{x}_j)$  can be used in the search for the separable vector  $\mathbf{a}$  [8].

The line  $y(\mathbf{w}) = \mathbf{w}^T \mathbf{x}$  can be designed on the basis of differential vectors  $\mathbf{r}_{m,n} = (\mathbf{x}_m - \mathbf{x}_n)$  from the learning sets  $C_k$  (1) with condition (8) through minimization of the convex and piecewise linear (*CPL*) criterion function  $\Phi(\mathbf{w})$  [8]:

$$(\forall k, k' \in \{1, \dots, K\}, \text{ where } k < k') (\forall \mathbf{x}_n \in C_k) (\forall \mathbf{x}_m \in C_{k'}) \quad \mathbf{r}_{m,n} = \mathbf{x}_m - \mathbf{x}_n \tag{20}$$

The *CPL* penalty functions  $\varphi_{m,n}(\mathbf{w})$  (17) can be defined on the basis of differential vectors  $\mathbf{r}_{mn}$ :

$$\begin{aligned} (\forall \mathbf{r}_{m,n}(20)) \quad \varphi_{m,n}(\mathbf{w}) &= 1 - (\mathbf{w})^T \mathbf{r}_{m,n} \quad \text{if } (\mathbf{w})^T \mathbf{r}_{m,n} < 1 \quad \text{or} \\ \varphi_{m,n}(\mathbf{w}) &= 0 \quad \text{if } (\mathbf{w})^T \mathbf{r}_{m,n} \geq 1 \end{aligned} \tag{21}$$

The *CPL* criterion function  $\Phi_d(\mathbf{w})$  is defined similarly to (18) [8]:

$$\Phi_d(\mathbf{w}) = \sum_{m,n} \alpha_{m,n} \varphi_{m,n}(\mathbf{w}) \tag{22}$$

where  $\alpha_{m,n}$  ( $\alpha_{m,n} > 0$ ) are positive parameters (*prices*). We are taking into account the minimal value:

$$\Phi_d^* = \Phi(\mathbf{w}_d^*) = \min_{\mathbf{w}} \Phi(\mathbf{w}) \geq 0 \tag{23}$$

**Lemma 3.** *If the learning sets  $C_k$  (1) fulfill the conditions (8) for unknown parameter vector  $\mathbf{a}$  then the minimal value  $\Phi_d^*$ , of the criterion function  $\Phi_d(\mathbf{w})$  (23) defined on the differential vectors  $\mathbf{r}_{m,n} = (\mathbf{x}_m - \mathbf{x}_n)$  is equal to zero ( $\Phi_d^* = 0$ ) and all the feature vectors  $\mathbf{x}_j(k)$  from each set  $C_k$  (1) are contained in the layer (slice)  $L(\mathbf{w}_d^*, \tau_k, \tau_{k+1})$  (9).*

As it results from the Lemma 2, the optimal vector  $\mathbf{w}_d^*$  constituting the minimum (22) of the criterion function  $\Phi_d(\mathbf{w})$  can be used as an estimator of an unknown model vector  $\mathbf{a}_1$  ( $\mathbf{a}_1 \approx \mathbf{w}_d^*$ ) (8). The linear separability (3) in the pairs  $\{C_k, C_l, k \neq l\}$  is preserved on the line  $\mathbf{l}(\mathbf{w}_d^*)$  (4).

Let us now regard the projections (7) on the plane  $\mathbf{P}(\mathbf{w}_1, \mathbf{w}_2)$  (5) defined by two parameter vectors  $\mathbf{w}_1$  and  $\mathbf{w}_2$ . We will assume the conditions (11) of the *Model 2* and will take into consideration the projection designing developed in the "Hepar" system of medical diagnosis support [10, 11].

We will use the projection plane  $\mathbf{P}(\mathbf{w}_1^*, \mathbf{w}_2^*)$  (5) defined by two optimal parameter vectors  $\mathbf{v}_i^* = [-\theta_i^*, (\mathbf{w}_i^*)^T]^T$  ( $i = 1, 2$ ) constituting minimum (22) of two criterion functions  $\Phi_i(\mathbf{v})$  (18). The functions  $\Phi_1(\mathbf{v})$  and  $\Phi_2(\mathbf{v})$  are specified separately for two axes of the projection plane. Each function  $\Phi_i(\mathbf{v})$  ( $i = 1, 2$ ) should be specified by the positive  $G_i^+$  and the negative  $G_i^-$  sets (14) of the feature vectors  $\mathbf{x}_j$  (1).

EXAMPLE: Let us consider four learning sets  $C_1, C_2, C_3, C_4$  (1) and below sets  $G_i^+$  and  $G_i^-$  (14):

$$\begin{aligned} G_1^+ &= C_2 \cup C_3 \quad \text{and} \quad G_1^- = C_1 \cup C_4 \\ G_2^+ &= C_1 \cup C_2 \quad \text{and} \quad G_2^- = C_3 \cup C_4 \end{aligned} \tag{24}$$

**Lemma 4.** *If the sets  $G_i^+$  and  $G_i^-$  (11) are linearly separable (3) for each of two axes, then the visualizing plane  $\mathbf{P}(\mathbf{w}_1^*, \mathbf{w}_2^*)$  (5) defined by two optimal parameter vectors  $\mathbf{w}_1^*$  and  $\mathbf{w}_2^*$  (19) has the below structure:*

$$\begin{aligned} \text{the upper - left quarter} &- C_1, & \text{the upper - right quarter} &- C_2 \\ \text{the lower - right quarter} &- C_3, & \text{the lower - left quarter} &- C_4. \end{aligned} \tag{25}$$

The above structure means among others that all the projected vectors  $\mathbf{x}_j$  (1) from the learning set  $C_1^-$  (1) are located as the points  $\mathbf{y}'_j$  (1) (7) in the upper-left quarter of the visualizing plane.

A similar Lemma can be formulated also for the sets  $G_i^+$  and  $G_i^-$  different from (24).

On the basis of the structure of the points  $\mathbf{y}'_j(k)$  (7) on the visualizing plane  $\mathbf{P}(\mathbf{w}_1^*, \mathbf{w}_2^*)$  we can infer about structure of the sets  $G_1^+, G_1^-, G_2^+, G_2^-$ . For example, if the visualizing plane  $\mathbf{P}(\mathbf{w}_1^*, \mathbf{w}_2^*)$  (7) has the structure (25) then adequate sets  $G_i^+$  and  $G_i^-$  (24) are linearly separable (12) in the feature space  $\mathbf{X}[n]$  for  $i = 1, 2$ . Such conclusion is an example of the *inverse inference*. The *inverse inference* allows for a search for interesting patterns in the feature space  $\mathbf{X}[n]$  on the basis of the visualized points  $\mathbf{y}'_j(k)$  (7).

## 7 Concluding Remarks

Separable transformations of the learning sets  $C_k$  (1) allow for induction of the similarity measure or distance function for the *K-NN* rule or in the *Case Based Reasoning* scheme. It has been shown experimentally within the system "Hepar" that the separable projections of multivariate data sets ( $n \approx 50$ ) on the visualizing plane (7) allowed to improve the classification accuracy of the *K-NN* rules used in medical diagnosis support [11].

An inspection of the visualizing plane (7) can allow also to identify potentially useful patterns (regularities) in the feature space  $\mathbf{X}[n]$ . For example, an inverse inference based on the *Lemma 4* can be used for a search of such data subsets which are linearly separable (12) in the feature space  $\mathbf{X}[n]$ .

## Acknowledgment

This work was partially supported by the grants KBN 3T11F011 30, by the grant W/WI/1/2005 from the Białystok Technical University.

## References

1. Johnson RA, Wichern DW (1991) Applied Multivariate Statistical Analysis, Prentice-Hall Inc., Englewood Cliffs, New York, 1991
2. Duda OR, Hart PE, Stork DG (1986) Pattern Classification, J. Wiley, New York
3. Jolliffe I (1986) Principal Component Analysis. Springer-Verlag, New York, 1986
4. Lachenbruch P (1975) Discriminant analysis, New York: Hafner Press, first edition, 1975
5. Friedman JH, Tukey JW (1974) A projection pursuit algorithm for exploratory data analysis, IEEE Trans. of Computers, c-23(9):881-890, 1974.
6. Bobrowski L (2005) Exploratory data analysis based on convex and piecewise linear (CPL) criterion functions (in Polish), Białystok Technical University, 2005
7. Klein JP, Moeschberger ML (1997) Survival Analysis, Techniques for Censored and Truncated Data, Springer, NY 1997
8. Bobrowski L (2005) Ranked modeling with feature selection based on the CPL criterion functions, in Machine Learning and Data Mining in Pattern Recognition, Eds. P. Permer et al., Lecture Notes in Computer Science vol. 3587, Springer Verlag, Berlin 2005
9. Bobrowski L (1991) Design of piecewise linear classifiers from formal neurons by some basis exchange technique, Pattern Recognition, 24(9), pp. 863-870, 1991.
10. Bobrowski L, Wasyluk H, (2001) Diagnosis supporting rules of the Hepar system, pp. 1309-1313 in: MEDINFO 2001, Proceedings of the 10-th World Congress on Medical Informatics, Ed. by V. L. Patel, R. Rogers, R Haux, IMIA, IOS Press Amsterdam 2001
11. Bobrowski L, Wasyluk H (2007) Induction of Diagnostic Support Rules through Data Mapping - on the Example of the Hepar system, Biocybernetics and Biomedical Engineering (in press)

---

# Support Vector Machine Classifier with Feature Extraction Stage as an Efficient Tool for Atrial Fibrillation Detection Improvement

Paweł Kostka and Ewaryst Tkacz

Institute of Electronics, Division of Microelectronics and Biotechnology, Silesian University of Technology, Gliwice, Poland  
{pkostka,etkacz}@polsl.pl

**Summary.** Intensively explored support vector machine (SVM) due to its several unique advantages was successfully verified as a time series predicting tool in last years. In presented work the improvement of SVM classifier by introduction to a system a preliminary feature extraction stage is proposed. Based on ECG signals from patients suffering from atrial fibrillation (AF) a new feature vector based on separate time, frequency and mixed-domain time-frequency (TF) parameters was created. As a efficient tool for non-stationary signals analysis the discrete wavelet transform was used to obtain the TF signal representation and then new parameters based on energy and entropy measure were computed. Proposed system (FESVM) was tested on the set of 20 AF and 20 patients from control group (CG) divided into learning and verifying subsets. Obtained results showed, that the ability of generalization for enriched FESVM based system increased, due to selectively choosing only the most representative features for analyzed AF detection problem.

## 1 Introduction

Significant advantages of Support Vector Machine (SVM) [1], such as good generalization performance, the absence of local minima and the sparse representation of solution are the reason for treating SVM as a powerful tool in classifier systems. SVM learning systems are originated in modern statistical learning theory and found a wide range of real-world application in e.g. object detection [2] and face [3] or text recognition [4]. Based on principle of structural risk minimization (rather than empirical risk minimization like in other methods) it tries to minimize an upper bound of the generalization error rather than minimize the training error (like e.g. MSE algorithm based on training samples). This algorithm focuses on training 'borderline' cases, just called 'support vectors', that are the most difficult to classify by maximization the margin between classes. As the training of SVM is equivalent to solving a linearly constrained convex quadratic programming problem, the solution of SVM is always global optimal and absent from local minima. Because this method is sensitive for irrelevant or correlated inputs, which can decrease its generalization performance the feature extraction or selection stage is very recommended. In proposed structure features are extracted based on ECG analysis in time (T), frequency (F) and

mixed time-frequency (TF) domains. The main task for feature extraction and selection stage of biomedical signal semi-automatic classifiers is to reduce too high size of input feature space and reveal only discriminate features for given task. The application field of presented SVM based classifier with multi-domain feature extraction is the trial of detection of atrial fibrillation (AF), which is a supraventricular tachyarrhythmia, characterized by uncoordinated atrial activation with consequent deterioration of atrial mechanical function. Last researches report AF as a result of a fractionated atrial electrical activity mainly due to the shortening of atrial refractory period, which allows multiple waves pass through the atrial mass. These changes ultimately reduce the inward calcium current, and this in turn reduces the action potential duration. If the action potential duration shortens, the refractory period shortens too, and the cell can be ready for reactivation earlier [5] On the electrocardiogram (ECG), AF is described by the replacement of consistent P waves by rapid oscillations or fibrillatory waves that vary in size, shape, and timing, associated with an irregular, frequently rapid ventricular response when atrioventricular conduction is intact. Because of disturbed hemodynamic, atrial fibrillation and atrial flutter are between of the most usual causes of thrombi-embolic events [5]. So it is very important to diagnose and to treat these arrhythmias.

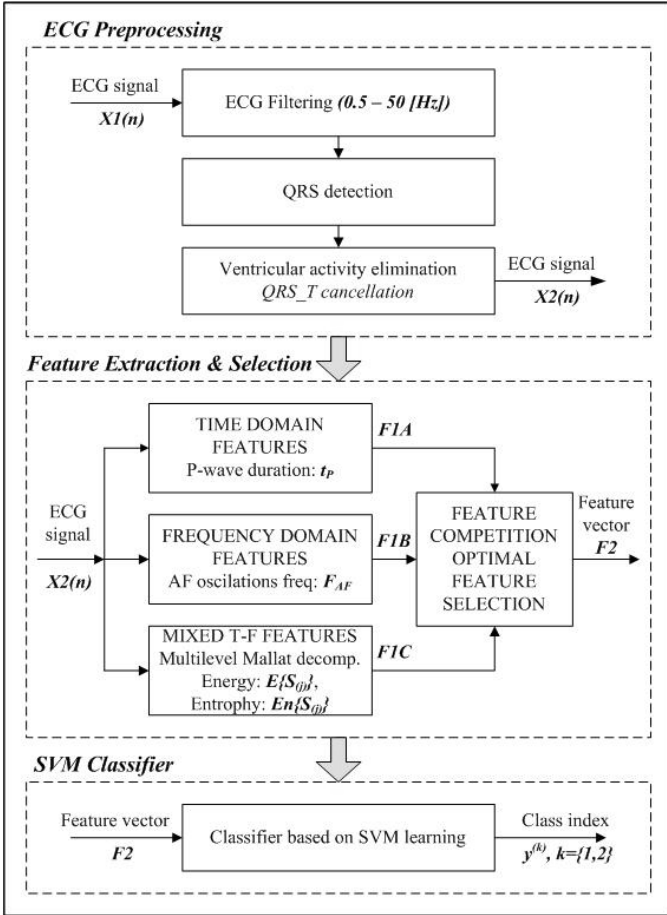
## 2 Methods

### 2.1 General Structure

General structure of described classifier system for screening examination of patients suffered from atrial fibrillation, presented in (figure(1)) includes feature extraction stage with proposed mixed-domain structure, following by SVM classifier structure with preliminary original ECG signal preprocessing with important ventricular activation (QRST) cancelation stage.

### 2.2 Ventricular Activity Cancelation

Literature review of papers connected to AF detection problem shows positive influence of ventricular activation cancelation by removing QRST complex from original signal for further analysis [6]. Traditional methods based on simple QRS cycles averaging and subtraction was not enough in case of significant beat-to-beat changes in real ECG signal, which can not be treated as periodical signal. Recent studies report principal component analysis, blind source separation and artificial neural networks [7] as a alternative promising tools for QRST complex cancelation. In presented approach after limiting the ECG bandwidth to the range  $(0.5 \div 50 [Hz])$ , we proposed to use the advantages of discrete wavelet transform analysis dedicated for non-stationary signals implement as the threshold reconstruction from details  $d4 + d5$  multilevel Mallat decomposition components.



**Fig. 1.** General structure of SVM classifier with preliminary feature extraction on ECG signal after QRST cancellation

### 2.3 Feature Extraction

#### Time Domain Parameters

For P wave duration  $t_P$  measurement, the crucial problem consists in the definition of the fiducial points [8]. Starting and ending point of the P wave can be detected either manually or automatically. In some cases they are defined as the points corresponding respectively to the first (onset) and last (off-set) deflection from the baseline. In our approach the results of Mallat discrete wavelet transform was used and the sum of d6+d7 details corresponding to atrial activity, after VA cancellation was the base for  $t_P$  estimation as a time when its absolute value increases given threshold  $tr_1$  (between two consecutive QRS complexes).

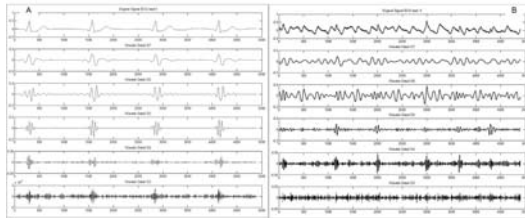
### Frequency Domain Parameters

Frequency localization of atrial activation in power spectrum density (PSD), calculated from ECG signal after QRST cancelation was the source for  $F_P$  parameter estimation.

### Time-Frequency Signal Analysis

As a feature extraction tool the wavelet transform based on multilevel Mallat signal decomposition [9], [10] was used. Taking into consideration specific features of the ECG signal, especially default localization of its significant frequency components the grid of discrete wavelet scale - a values was created, corresponding to Mallat signal decomposition levels (for sampling frequency  $f_S = 250[Hz]$ , five levels corresponding to scale values:  $a_i, i = 3..7$  were taken into consideration)(see figure(2)for comparison of Mallat decomposition for CG (A) and AF (B) case). As a next step, to create the new features vector, for every signal component obtained on each decomposition level a set of parameters was computed. For each subspace wavelet coefficients were squared and normalized to obtain the energy probability distribution (3)

$$p_i = \frac{c_i^2}{\sum c_k^2} \tag{1}$$



**Fig. 2.** Discrete wavelet transform by Mallat decomposition of II lead ECG signal from normal CG (A) and AF (B) patient

For each wavelet scale, the sorted series may be considered as an inverse empirical cumulative energy distribution function (ECDF). Base on this parameters the Shannon entropy  $E$  (4) of energy distribution  $p_i$  (3) were calculated as a measure of energy unpredictability in each wavelet decomposition subspace.

$$E = \sum p_i \log_2(p_i) \tag{2}$$

This procedure allowed to reveal a group of new features based on energy and entropy measures. The whole set of new feature vectors  $\mathbf{F1}_{C1}.. \mathbf{F1}_{C5}$ , created as a result of multilevel Mallat signal decomposition, which is put to the input of classifier structure includes the following groups of parameters as a series for every of  $i^{th}$  decomposition level:

- Mean values of wavelet coefficients in each sub-band (frequency distribution information)
- Standard deviations of wavelet coefficients (level of change of frequency distribution information)
- Energy of  $i^{th}$  component (3)
- Shannon entropy of wavelet component (distribution of the amount of information included in every sub-band)
- Shannon entropy E of energy distribution  $p_i$

### 2.4 Classifier Stage - Support Vector Machine Learning System

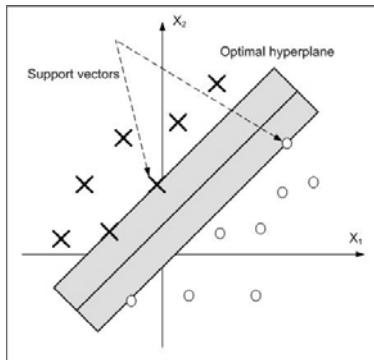
For real world application of SVM with additional feature extraction stage (FESVM) the atrial fibrillation detection was chosen as a two-class pattern recognition problem. For the input pattern (feature vector)  $\mathbf{x} \in X \subseteq \mathbb{R}^n$  and the scalar  $y$  denoting its class label (in our case:  $y \in \{\pm 1\}$ ) let  $(\mathbf{x}_i, y_i), i = 1, 2, \dots, l$  is a given set of  $l$  training samples.

#### Linear SVM Classifier

In case of linearly separable data, the goal of training phase of SVM is to find the linear function 3:

$$f(\mathbf{x}) = \mathbf{w}^T \times \mathbf{x} + b \tag{3}$$

which is the border for two different data classes. SVM classifier from infinite number of possible separate hyperplanes finds this one, which maximizes the margin between two classes Figure(3). Support vectors are the elements of training data set, which lies on the margin border. This optimal hyperplane can be found by minimizing following cost function 4:



**Fig. 3.** SVM classification scheme with hyperplane maximizing the margin between two classes. Support vectors are created from training set samples lying on the margin border.



$$J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} = \frac{1}{2} \|\mathbf{w}\|^2 \tag{4}$$

In real, specific problem, in case, when training data may not be completely separable by a hyperplane an additional variable  $\epsilon_i$  may be introduced to soften the stiff hyperplane constrains. The cost function to minimize is modified in following way 5:

$$J(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \epsilon_i \tag{5}$$

where  $C$  is a user-specified, positive regularization parameter and  $\epsilon$  is a vector containing constants for all  $l$  training data pairs.

**Nonlinear SVM Classifier**

Extension of SVM to nonlinear classification can be done by introduction the nonlinear operator  $\Phi(\bullet)$ , which maps the input vector  $\mathbf{x}$  into a higher dimension space  $\aleph$ , where is possible to perform procedure of margin between classes maximization. In this form the nonlinear SVM classifier is defined as 6:

$$f(\mathbf{x}) = \mathbf{w}^T \times \Phi(\bullet) + b \tag{6}$$

Analogues the parameters of decision function 6 are obtained by minimization of modified cost function 7:

$$\min J(\mathbf{w}, \epsilon) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \epsilon_i \tag{7}$$

To minimize  $J(\mathbf{w}, \epsilon)$  a Lagrange multipliers can be used, what lead to the formula expressed vector  $\mathbf{w}$  as a linear combination of mapped vectors  $\Phi(x_i)$  8:

$$vecw = \sum_{i=1}^l \alpha_i y_i \Phi(x_i) \tag{8}$$

where  $\alpha_i \geq 0, i = 1, 2, \dots, l$  are the Lagrange multipliers.

The nonlinear SVM kernel function  $\Phi(\bullet)$  is used to map by inner product operation input SVM vector to high-dimensional feature space. From common choises used in SVM research Gaussian radial basis 9 and polynomial 10 kernel was tested.

$$K(x, y) = (x^T y + 1)^p \tag{9}$$

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \tag{10}$$

Two mentioned above kernel functions were compared in section III (Results).

**3 Results**

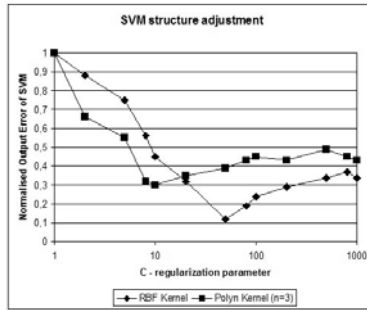
We tested the performance of our algorithm on the MITBIH Atrial Fibrillation (AF) and normal sinus rhythm (control group CG) database. AF database contains 10 hour ambulatory recordings from 40 patients using 2-lead electrodes in a standard configuration and 40 CG patients. The data set was divided equally into learning and verifying set.

### 3.1 Feature Extraction Stage Adjustment

After extraction of 32 element’s feature set from ECG signal (see Section II) using our results of feature selection algorithm based on Discriminate Measure estimation described in our previous work [11] set of 9 the most sensitive features for AF detection was chosen.

### 3.2 SVM Classifier Structure Optimization

For empirically chosen parameters of both polynomial (order  $n$ ) and Gaussian RBF ( $\sigma$ ) nonlinear kernel, the influence of the value of SVM nonlinear mapping regularization parameter  $C$  was tested. (Figure(4)) presents the normalized value of generalized classification error of SVM structure with feature extraction layer, for two mentioned kernel functions as the function of  $C$  parameter.



**Fig. 4.** C regularization param. of SVM structure adjustment for two nonlinear kernel function

**Table 1.** Verification of SVM classifier with two kernel functions

SVM structure type	Sensitivity (SN)	Specificity (SP)
SVM without FE (Polyn. kernel)	0.72	0.69
SVM without FE (RBF. kernel)	0.76	0.70
SVM + FE (Polyn. kernel)	0.84	0.82
SVM + FE (RBF. kernel)	0.88	0.86

### 3.3 FESVM Classifier Verification

The main significant feature characterizing classifier performance is its generalization ability. The measure of Sensitivity (SN) and Specificity (SP) was calculated for the chosen in previous subsection structure of whole SVM classifier with feature extraction stage (FESVM). Apart from classification performance Table (1) presents also the influence of proposed feature extraction stage for the values of SN and SP values obtained for the SVM classifier. Presented results

were calculated for two kernel functions of nonlinear SVM (Gaussian RBF and  $3^{rd}$  order polynomial) with and without feature extraction layer.

## 4 Discussion and Conclusion

Numerical experiments allowed to state, that the structure of SVM based classifier with the feature extraction (FE) stage is very sensitive for both SVM adjustment (C parameter and kernel type) and input features set creation (Figure 4). Table 1 presents the positive influence of proposed feature extraction preliminary stage on SVM performance for atrial fibrillation detection problem.

## References

1. Vapnik V., *Statistical Learning Theory*, Wiley, New York, 1998.
2. Pontil M. Verri A., Support vector machine for 3-D object recognition, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.20, pp.637-646, June 1998.
3. Osuna E. Freund R. Girosi F., Training support vector machines: Application to face detection, in *Proc. Computer Vision and Pattern Recognition*, pp.130-136, Puerto Rico, 1997.
4. Joachims T., Transductive inference for text classification using support vector machines, *Proc. Int. Conference of Machine Learning*, Slovenia, June 1999.
5. Khand A.U., Rankin A.C., Vaye G.C. Systematic review of the management of atrial fibrillation in patients with heart failure. *Eur. Heart J*, Vol. 21, pp.614-632, 2000.
6. Sanchez C., Millet J., Rieta J. J., Rodenas J., Castells F., Ruiz R., and Ruiz V., "Packet Wavelet Decomposition: An Approach For Atrial Activity Extraction", *IEEE Computers in Cardiology*, vol.29 Sept.2002
7. Vasquez C., Hernandez A., Mora F., Carrault G., and Passariello G., 'Atrial activity enhancement by Wiener filtering using an artificial neural network', *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 8, pp. 940-944, Aug.2001.
8. Buxton AE, Josephson ME., The role of P wave duration as a predictor of post-operative atrial arrhythmias,80(68-73) *Chest*, July 1981.
9. Mallat S., A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transaction on Pattern Analysis*, 7(11):674-693, 1989.
10. Akay M., Time-Frequency and wavelet analysis, *IEEE EMB Magazine* 14(2), 1995.
11. Kostka P. Tkacz E., Hybrid Feature Vector Creation for Atrial Fibrillation Detection Improvement, *Proc. World Congress of Medical Physics and Biomedical Engineering*, Seoul, 4 pages, Sept 2006.

---

# Classification Accuracy in Local Optimal Strategy of Multistage Recognition with Fuzzy Data

Robert Burduk

Chair of Systems and Computer Networks, Wrocław University of Technology,  
Wybrzeże Wyspińskiego 27, 50-370 Wrocław, Poland  
robert.burduk@pwr.wroc.pl

**Summary.** The paper deals with the multistage recognition task. In this problem of recognition the Bayesian statistic is applied. The decision rules minimize the mean risk, that is the mean value of the zero-one loss function. The information on objects features is fuzzy or non-fuzzy. The probability of misclassification for local optimal strategy and the difference between probability of misclassification for the both information's are presented. Simple example of this difference conclude the work.

## 1 Introduction

This paper describes hierarchical classifier problem which is characterized by the fact, that an unknown pattern is classified into a class using several decisions in a successive manner. The mechanics of multistage classification can be described by means of a tree, in which the terminal nodes represent labels of classes and the interior nodes denote the groups of classes. Obviously, the multistage recognition scheme offers much more flexibility than the simple single stage classifier operating with only one set of features and basing its decision on the maximum selection among a set of discriminant values which are calculated independently of each other. However, this flexibility with respect to the tree structure the numerous parameters necessary to define in detail the decision rules at the different nodes turns out to be a major obstacle to the systematic design of a decision tree.

Classical statistical techniques assume that both the data and the probability functions are represented by known numerical values thus knowledge of data is precise. In practice, we are often faced with two types of uncertainty – randomness and fuzziness. Randomness is described and investigated using probability theory methods which satisfy statistical laws. In this type of uncertainty subjective influences are not taken into account. Fuzziness is characterized by nonstatistical properties and subjective influences. It is based on the fuzzy set theory. There are many cases where the available information is a mixture of randomness and fuzziness. A simple example of such situation is classification where the observations of the features are fuzzy-valued, but the probabilities of classes are numerical. In order to manage to treat both types of uncertainties, it

is necessary to incorporate fuzzy concept into statistical technique. This problem has been studied by many authors. Following papers describe statistical point estimation in the fuzzy environment [5, 15], fuzzy decision trees [8], testing of fuzzy hypotheses [4, 6], fuzzy Bayesian statistics [1, 14] and other combinations of statistical methods and fuzzy set theory.

In this paper, we consider the problem of probability of misclassification in multistage classifier when observations of the features are fuzzy-valued. Additionally, this paper describe probability of misclassification of local optimal strategy of multistage classification.

## 2 Bayesian Hierarchical Classifier

In the paper [9] the Bayesian hierarchical classifier is presented. The synthesis of multistage classifier is a complex problem. It involves specifying of the following components:

- the decision logic, i.e. hierarchical ordering of classes,
- feature used at each stage of decision,
- the decision rules (strategy) for performing the classification.

This paper is devoted only to the last problem. This means that we will deal only with the presentation of decision algorithms, assuming that both the tree skeleton and the feature used at each non-terminal node have been specified.

The procedure in the Bayesian hierarchical classifier consists of the following sequences of activities. At the first stage, there are measured some specific features  $x_0$ . They are chosen from among all accessible features  $x$ , which describe the pattern that will be classified. These data constitute a basis for making a decision  $i_1$ . This decision, being the result of recognition at the first stage, defines a certain subset in the set of all classes and simultaneously indicates features  $x_{i_1}$  (from among  $x$ ) which should be measured in order to make a decision at the next stage. Now at the second stage, features  $x_{i_1}$  are measured, which together with  $i_1$  are a basis for making the next decision  $x_2$ . This decision – like  $i_1$  – indicates features  $x_{i_2}$  necessary to make the next decision (at the third stage) and – like at the previous stage – defines a certain subset of classes, not in the set of all classes, however, but in the subset indicated by the decision  $i_1$ , and so on. The whole procedure ends at the last  $N$ -th stage, where the decision made  $i_N$  indicates a single class, which is the final result of multistage recognition. Thus multistage recognition means a successive narrowing of the set of potential classes from stage to stage, down to a single class, simultaneously indicating at every stage features which should be measured to make the next decision in more precise manner.

## 3 Decision Problem Statement

Let us consider a pattern recognition problem, in which the number of classes equals  $M$ . Let us assume that the classes are organized in a  $(N + 1)$  horizontal

decision tree. Let us number all the nodes of the constructed decision-tree with consecutive numbers of  $0, 1, 2, \dots$ , reserving 0 for the root-node and let us assign numbers of classes from the  $\mathcal{M} = \{1, 2, \dots, M\}$  set to terminal nodes so that each one of them can be labelled with the number of the class connected with that node. This allows us to introduce the following notation:

- $\mathcal{M}(n)$  – the set of numbers of nodes, which distance from the root is  $n$ ,  $n = 0, 1, 2, \dots, N$ . In particular  $\mathcal{M}(0) = \{0\}$ ,  $\mathcal{M}(N) = \mathcal{M}$ ,
- $\overline{\mathcal{M}} = \bigcup_{n=0}^{N-1} \mathcal{M}(n)$  – the set of interior node numbers (non terminal),
- $\mathcal{M}_i \subseteq \mathcal{M}(N)$  – the set of class labels attainable from the  $i$ -th node ( $i \in \overline{\mathcal{M}}$ ),
- $\mathcal{M}^i$  – the set of numbers of immediate descendant nodes ( $i \in \overline{\mathcal{M}}$ ),
- $m_i$  – number of direct predecessor of the  $i$ -th node ( $i \neq 0$ ).

We will continue to adopt the probabilistic model of the recognition problem, i.e. we will assume that the class label of the pattern being recognized  $j_N \in \mathcal{M}(N)$  and its observed features  $x$  are realizations of a couple of random variables  $\mathbf{J}_N$  and  $\mathbf{X}$ . The complete probabilistic information denotes the knowledge of a priori probabilities of classes:

$$p(j_N) = P(\mathbf{J}_N = j_N), \quad j_N \in \mathcal{M}(N) \tag{1}$$

and class-conditional probability density functions:

$$f_{j_N}(x) = f(x/j_N), \quad x \in X, \quad j_N \in \mathcal{M}(N). \tag{2}$$

Let

$$x_i \in X_i \subseteq R^{d_i}, \quad d_i \leq d, \quad i \in \mathcal{M} \tag{3}$$

denote vector of features used at the  $i$ -th node, which have been selected from the vector  $x$ .

Our aim is now to calculate the so-called multistage recognition strategy  $\pi_N = \{\Psi_i\}_{i \in \overline{\mathcal{M}}}$ , that is the set of recognition algorithms in the form:

$$\Psi_i : X_i \rightarrow \mathcal{M}^i, \quad i \in \overline{\mathcal{M}}. \tag{4}$$

Formula (4) is a decision rule (recognition algorithm) used at the  $i$ -th node which maps observation subspace to the set of immediate descendant nodes of the  $i$ -th node. Analogically, decision rule (4) partitions observation subspace  $X_i$  into disjoint decision regions  $D_{x_i}^k$ ,  $k \in \mathcal{M}^i$ , such that observation  $x_i$  is allocated to the node  $k$  if  $k_i \in D_{x_i}^k$ , namely:

$$D_{x_i}^k = \{x_i \in X_i : \Psi_i(x_i) = k\}, \quad k \in \mathcal{M}^i, \quad i \in \overline{\mathcal{M}}. \tag{5}$$

Our aim is to minimize the mean risk function (the probability of misclassification) denoted by:

$$\overline{R}(\overline{\pi}_N^*) = \min_{\Psi_{i_n}, \dots, \Psi_{i_{N-1}}} R(\pi_N) = \min_{\Psi_{i_n}, \dots, \Psi_{i_{N-1}}} E[L(I_N, J_N)]. \tag{6}$$

The  $\bar{\pi}_N^*$  strategy we will call the locally optimal  $N$ -stage recognition strategy. Formally, this strategy can be derived minimizing the local criteria, which denote probabilities of misclassification for particular nodes of a tree. Its decision rules are mutually independent. In the next chapter we will present difference between probability of misclassification in locally optimal strategy for the fuzzy and non-fuzzy information on objects features. Consideration will be limited to the case of zero-one loss function.

### 4 Classification Accuracy

The recognition algorithm at the  $n^{\text{th}}$  stage for crisp data is as follows [10]:

$$\begin{aligned} \bar{\Psi}_{i_n}(\tilde{A}_{i_n}) &= i_{n+1} \quad \text{when} \\ p(i_{n+1})f_{i_{n+1}}(x_{i_n}) &= \max_{k \in \mathcal{M}^{i_n}} p(k)f_k(x_{i_n}). \end{aligned} \tag{7}$$

A fuzzy information  $\tilde{A}_i$  from  $X_i$  is a fuzzy event characterized by membership function  $\mu_{\tilde{A}_i}(x_i)$ . Fuzzy observation is described by fuzzy number or fuzzy trapezoidal interval [11] represented by the following simple notation: fuzzy triangular number –  $\tilde{A} = (a_1, a_2, a_3, a_4)$ , fuzzy trapezoidal interval –  $\tilde{A} = (a_1, a_2, a_3, a_4)_{Tr}$ . In fuzzy triangular number  $a_2 = a_3$ . In addition, assume that for each component  $k$  of observation subspace  $X_i$  the set of all available fuzzy numbers  $\tilde{\mathcal{A}}_i = \{\tilde{A}_i^1, \tilde{A}_i^2, \dots, \tilde{A}_i^k\}$  satisfies the orthogonality constraint [12]. Probability of fuzzy event assume in Zadeh’s form [16]:

$$P(\tilde{A}) = \int_{supp(\tilde{A})} \mu_{\tilde{A}}(x)f(x)dx, \tag{8}$$

where  $supp(\tilde{A})$  denotes carrier of the  $\tilde{A}$  set.

Applying procedure similar to [2] and use zero-one loss function we obtain searched locally optimal strategy with decision algorithms as follows:

$$\begin{aligned} \bar{\Psi}_{i_n}(\tilde{A}_{i_n}) &= i_{n+1} \quad \text{when} \\ p(i_{n+1}) \int_{supp \tilde{A}_{i_n}} \mu_{\tilde{A}_{i_n}}(x_{i_n})f_{i_{n+1}}(x_{i_n})dx_{i_n} &= \\ = \max_{k \in \mathcal{M}^{i_n}} p(k) \int_{supp \tilde{A}_{i_n}} \mu_{\tilde{A}_{i_n}}(x_{i_n})f_k(x_{i_n})dx_{i_n}. \end{aligned} \tag{9}$$

When we use fuzzy information on object features instead of exact information we deteriorate the classification accuracy.

**Theorem 1.** *The difference between probability of misclassification for the both fuzzy data and crisp data information’s in individual node  $i_n$  is the following:*

$$Pe_F(i_n) - Pe(i_n) = \tag{10}$$

$$= \sum_{\tilde{A}_{i_n} \in X_{i_n}} \left( \int_{\text{supp } \tilde{A}_{i_n}} \mu_{\tilde{A}_{i_n}}(x_{i_n}) \max_{j_{n+1} \in \mathcal{M}^{i_n}} \left\{ \frac{p(j_{n+1})}{p(i_n)} f_{j_{n+1}}(x_{i_n}) \right\} dx_{i_n} - \right. \\ \left. - \max_{j_{n+1} \in \mathcal{M}^{i_n}} \in \mathcal{M}^{i_n} \left\{ \frac{p(j_{n+1})}{p(i_n)} \int_{\text{supp } \tilde{A}_{i_n}} \mu_{\tilde{A}_{i_n}}(x_{i_n}) f_{j_{n+1}}(x_{i_n}) dx_{i_n} \right\} \right),$$

where  $Pe_F(i_n)$  the probability of misclassification for fuzzy-data and  $Pe(i_n)$  for crisp data in the node  $i_n$ .

*Proof.* For the assumption 8 we have:

$$Pe_F(i_n) = 1 - \sum_{j_{n+1} \in \mathcal{M}^{i_n}} \frac{p(j_{n+1})}{p(i_n)} \sum_{\tilde{A}_{i_n} \in D_{x_{i_n}}^{(j_{n+1})}} \int_{\text{supp } \tilde{A}_{i_n}} \mu_{\tilde{A}_{i_n}}(x_{i_n}) f_{j_{n+1}}(x_{i_n}) dx_{i_n} = \\ = 1 - \sum_{\tilde{A}_{i_n} \in X_{i_n}} \max_{j_{n+1} \in \mathcal{M}^{i_n}} \left\{ \frac{p(j_{n+1})}{p(i_n)} \int_{\text{supp } \tilde{A}_{i_n}} \mu_{\tilde{A}_{i_n}}(x_{i_n}) f_{j_{n+1}}(x_{i_n}) dx_{i_n} \right\}$$

and

$$Pe(i_n) = 1 - \sum_{j_{n+1} \in \mathcal{M}^{i_n}} \frac{p(j_{n+1})}{p(i_n)} q(j_{n+1}/i_n, j_{n+1}) = \\ = 1 - \int_{X_{i_n}} \max_{j_{n+1} \in \mathcal{M}^{i_n}} \left\{ \frac{p(j_{n+1})}{p(i_n)} f_{j_{n+1}}(x_{i_n}) \right\} dx_{i_n} = \\ = 1 - \sum_{\tilde{A}_{i_n} \in X_{i_n}} \int_{\text{supp } \tilde{A}_{i_n}} \mu_{\tilde{A}_{i_n}}(x_{i_n}) \max_{j_{n+1} \in \mathcal{M}^{i_n}} \left\{ \frac{p(j_{n+1})}{p(i_n)} f_{j_{n+1}}(x_{i_n}) \right\} dx_{i_n}. \quad \square$$

Let us illustrate the obtained results with the following example.

### 5 Illustrative Example

Let us consider the two-stage binary classifier presented in Fig. 1. Four classes have identical a priori probabilities which equal 0.25. We use 3-dimensional data  $x = [x^{(1)}, x^{(2)}, x^{(3)}]$  where class-conditional probability density functions are normally distributed. For performing the classification at the root-node 0 the first coordinate was used, components  $x^{(2)}$  and  $x^{(3)}$  were used at the node 5 and 6 respectively. In the data covariance matrices are equal for every class  $\sum_{j_2} = 4I$ ,  $j_2 \in \mathcal{M}(2)$ , and the expected values are following  $\mu_1 = [0, 0, 0]$ ,  $\mu_2 = [0, 4, 0]$ ,  $\mu_3 = [3, 0, 1]$ ,  $\mu_4 = [3, 0, 8]$ . In experiments were used the following sets of fuzzy numbers:

Case A

$$\tilde{A} = \{ \tilde{A}_1 = (\infty, -9, -8), \tilde{A}_2 = (-9, -8, -8, -7), \dots, \\ \tilde{A}_{25} = (14, 15, 15, 16), \tilde{A}_{26} = (15, 16, \infty) \}.$$

Case B

$$\tilde{B} = \{ \tilde{B}_1 = (\infty, -9, -8.5), \tilde{B}_2 = (-9, -8.5, -8.5, -8), \dots, \\ \tilde{B}_{50} = (15, 15.5, 15.5, 16), \tilde{B}_{51} = (15.5, 16, \infty) \}.$$



Case C

$$\tilde{C} = \{\tilde{C}_1 = (\infty, -9.25, -8.75), \tilde{C}_2 = (-9.25, -8.75, -8.25, -7.75)_{Tr}, \dots, \tilde{C}_{26} = (14.75, 15.25, 15.75, 16.25)_{Tr}, \tilde{C}_{27} = (15.75, 16.25, \infty)\}.$$

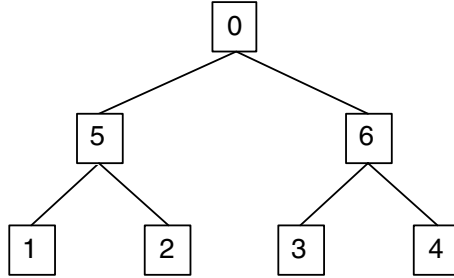


Fig. 1. Decision tree

Tab. 1 shows the difference between probability of misclassification for fuzzy and non fuzzy data in local optimal strategy of multistage classification calculated form (10).

Table 1. Difference of misclassification in each internal node

Case	$Pe_F(i_n) - Pe(i_n)$ in node		
	0	5	6
A	0,005	0,017	0,001
B	0,005	0,004	0,001
C	0,016	0,001	0,003

The classification error presented in Tab. 1 is calculated for a full probabilistic information.

## 6 Conclusions

In the present paper we have concentrated on the Bayesian statistics in the hierarchical classifier based on a decision tree scheme. Assuming a full probabilistic information, we presented the difference between probability of misclassification for fuzzy and crisp data. This difference is limited to the case for local optimal strategy of multistage classification. This strategy can be derived minimizing the local criteria, which denote probabilities of misclassification for particular nodes of a tree In [3] the upper bound of the difference between probability of misclassification for fuzzy and non fuzzy for global optimal strategy was presented.

Further research should concern combining techniques [7, 13] to improve of performance of multistage classifier with fuzzy observations.

**Acknowledgements.** This work is supported by The Polish State Committee for Scientific Research under the grant which is realizing in years 2006–2009.

## References

1. Berger J O (1985) *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, Berlin Heidelberg
2. Burduk R (2004) Decision Rules for Bayesian Hierarchical Classifier with Fuzzy Factor. *Soft Methodology and Random Information Systems, Advances in Soft Computing*. 519–526
3. Burduk R (2005) Probability of Misclassification in Bayesian Hierarchical Classifier. *Soft Methodology and Random Information Systems, Advances in Soft Computing*. 341–348
4. Casals M R, Gil M A, Gil P (1986) On the Use of Zadeh's Probabilistic Definition for Testing Statistical Hypotheses from Fuzzy Information. *Fuzzy Sets and Systems* **20**, 175–190
5. Gertner G Z, Zhu H (1996) Bayesian Estimation in Forest Surveys when Samples or Priori Information Are Fuzzy. *Fuzzy Sets and Systems* **77**, 277–290
6. Grzegorzewski P (2000) Testing Statistical Hypotheses with Vague Data. *Fuzzy Sets and Systems* **112**, 501–510
7. Hung W L (2001) Bootstrap Method for Some Estimators Based on Fuzzy Data. *Fuzzy Sets and Systems* **119**, 337–341
8. Janickow C Z (1998) Fuzzy Decision Trees: Issues and Methods. *IEEE Trans. Systems, Man, and Cybernetics B: Cybernetics* **28**, 1–14
9. Kurzyński M (1983) Decision Rules for a Hierarchical Classifier. *Pattern Recognition Letters* **1**, 305–310
10. Kurzyński M (1988) On the Multistage Bayes Classifier. *Pattern Recognition* **21**, 355–365
11. Möller B, Beer M (2004) *Fuzzy Randomness*. Springer-Verlag, Berlin Heidelberg
12. Pedrycz W (1990) Fuzzy Sets in Pattern Recognition: Methodology and Methods. *Pattern Recognition* **23**, 121–146
13. Woźniak M (2004) Proposition of Boosting Algorithm for Probabilistic Decision Support System. *LNCS* **3036**, 675–678
14. Viertl R (1996) *Statistical Methods for Non-Precise Data*. CRC Press, Boca Raton
15. Yao J S, Hwang C M (1996) Point Estimation for the n Sizes of Random Sample with One Vague Data. *Fuzzy Sets and Systems* **80**, 205–215
16. Zadeh L A (1968) Probability Measures of Fuzzy Events. *Journal of Mathematical Analysis and Applications* **23**, 421–427

---

# Experiments on Classifiers Obtained Via Decision Tree Induction Methods with Different Attribute Acquisition Cost Limit

Wojciech Penar<sup>1</sup> and Michal Wozniak<sup>2</sup>

<sup>1</sup> Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Wybrzeże Wyspińskiego 27, 50-370 Wrocław, Poland  
wojciech.penar@pwr.wroc.pl

<sup>2</sup> Chair of Systems and Computer Networks, Wrocław University of Technology, Wybrzeże Wyspińskiego 27, 50-370 Wrocław, Poland  
michal.wozniak@pwr.wroc.pl

**Summary.** The paper presents idea of cost sensitive learning method for decision tree induction with fixed attribute acquisition cost limit. Properties of mentioned concept are established during computer experiments conducted on chosen databases from UCI Machine Learning Repository.

## 1 Introduction

During designing project of computer recognition systems the cost of their designing and exploitation plays the key role. The cost of exploitation can be considered as the expenses of incorrect diagnosis or expenses of feature value acquisition. The first problem is the typical problem for Bayes decision theory where we want to find the classifier for which the cost of misclassification is the lowest one [1]. Of course we have to know the lost function in advance which give as values of cost in case differences between classifier's decision and real classification. In our paper we are concentrating our attention on the second case where the cost depends on real expenses of feature values acquisition for decision making [2] [15]. Of course it could be measured by monetary units or time ones. The typical example of cost sensitive diagnosis is medical diagnosis where physician would like to balance the costs of various tests with the expected benefits or doctors have to make the diagnosis fast on the base on the low cost (fast measured) features because therapeutical action has to taken without delay. Let's note that for the many decision tasks there are no problems to make the high-quality medical decision on the base on the expensive medical tests. One of the main task of designing computer aided diagnosis is finding the balance between cost of their exploitation and qualities of diagnosis. As we stated the problem of cost-sensitive decision making arises frequently in medicine [7], industrial production process [14], robotics [11], technological diagnosis [3] and many others fields, for example for electronic equipment testing and real-time computer systems. Additionally for many of typical diagnosis systems we can not exceed the cost limit what means that the maximum cost (or time) connected with diagnosis is fixed. Our

paper addresses with the most popular inductive learning method [4] based on the top-down decision tree induction. In the paper we propose the cost sensitive modifications of decision tree induction algorithm which respects on one hand the cost of feature values acquisition and on the other hand the maximum cost limit. The content of the work is as follows. Section 2 provides idea of decision tree induction algorithm and the related works take into account cost during this process. Section 3 describes proposition of method. Results of the experimental investigations of the algorithms are presented. Last section concludes the paper.

## 2 Cost Sensitive Decision Tree Induction

This section discusses the methods used in this paper. Firstly we present the idea of the top-down decision tree induction algorithm given by Quinlan and we propose related modification of this algorithm which respect the cost of feature values acquisition.

### 2.1 C4.5

Algorithm C4.5 [10] is the modification of ID3[9] method generating decision tree. Therefore let us present the main idea of the ID3 below.

```

[1] Create a Root node for tree
[2] IF all examples are positive
[3]     THEN return the single node tree Root with label yes
           and return.
[4] IF all examples are negative
[5]     THEN return the single node tree Root with label no
           and return.
[6] IF set of attributes is empty
[7]     THEN return the single node tree Root with
           label = most common value of label in the set
           of examples and return
[8] Choose ,,the best'' attribute A from the set of attributes.
[9] FOR EACH possible value  $v_i$  of attribute
[10] 1. Add new tree branch bellow Root, corresponding
           to the test  $A=v_i$ .
[11] 2. Let  $E_{v_i}$  be the subset of set of examples that
           has value  $v_i$  for A.
[12] 3. IF  $E_{v_i}$  is empty
[13]     THEN bellow this new branches add a leaf node with
           label = most common value of label in the set
           of examples
[14]     ELSE below this new branch add new subtree and do
           this function recursive.
[15] END
[16] RETURN Root

```

**Fig. 1.** Pseudocode of decision tree induction algorithm

The central choice in the ID3 algorithm is selecting „the best” attribute to test at each node in the tree. The proposed algorithm uses the information gain that measures how well the given attribute separates the training examples according to the target classification. This measure bases on the Shanon’s entropy of set  $S$ :

$$Entropy(S) = \sum_{i=0}^M -p_i \log_2 p_i \tag{1}$$

where  $p_i$  is the proportin of  $S$  belonging to clas  $i$ . The Information Gain of an attribute  $A$  relative to the collection of examples  $S$ , is defined as

$$Gain(S, A) = Entropy(S) - \sum_{c \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v), \tag{2}$$

where  $values(A)$  is the set of all possible values for attribute  $A$  and  $S_v$  is the subset of  $S$  for which  $A = v$ . The C4.5 algorithm modifies ID3 that at the beginning the tree generation procedure does not use the whole set of examples.

### 2.2 EG2

The EG2 algorithm [8] is modification of decision tree induction algorithm where in the place of the Information Gain the Information Cost Function is proposed. For attribute  $A$  it is given by the following formula

$$ICF(S, A) = \frac{2^{Gain(S,A)}}{(Cost(A) - 1)^\omega}, \tag{3}$$

where  $Cost(A)$  is the acquisition cost of  $A$  attribute and  $\omega$  is the strength of the bias toward the lowest cost attributes. In case of  $\omega = 0$  the feature acquisition cost is ignored and  $ICF$  as the same features as  $Gain$  function. In case of  $\omega = 1$  mentioned cost plays the most important role.

### 2.3 CS-ID3

This cost sensitive modification of information gain is given in [11][12][13]

$$CS(S, A) = \frac{Gain(S, A)}{Cost(A)^2}. \tag{4}$$

### 2.4 IDX

The IDX [6] uses the following function

$$IDX(S, A) = \frac{Gain(S, A)}{Cost(A)}. \tag{5}$$

## 3 Propositions of the New Method

We propose the following modification of decision tree algorithm which respects the cost limit. We use the the Nunez’s proposition (3) with different values

instead of information gain function to determine which attribute has to be chosen for node creation. We add the input parameter „cost limit” and we propose to modify the 6th line of algorithm

[6] IF set of attributes is empty or cost of previous chosen attributes and cost of any remaining one exceed the cost limit

and the 8th line

[8] Choose "the best" attribute A from the set of attributes which cost added to the cost of previous chosen attributes does not exceed the cost limit

### 4 Experimental Investigations

The aim of the experiment was to compare errors and size of the decision tree classifiers obtained via C4.5 procedure which consider the classification (attribute acquisition) cost limit. The conditions of experiments are as follow: All experiments were made for different limits of cost and  $\omega$  values for EG2 modification. For experiments we chose pruned decision tree obtained via C4.5 [10]. For pruning the Rule Post Pruning method was used. All experiments were carried out using modified Quinlan's implementation of C4.5 and own software created in

Table 1. Please write your table caption here

database	number of examples	number of attributes	number of classes
1 pima-indians-diabetes	768	8	2
2 heart-disease	303	13	2
3 hepatitis	155	19	2
4 liver-disorders	345	5	2
5 thyroid-disease	7200	20	3

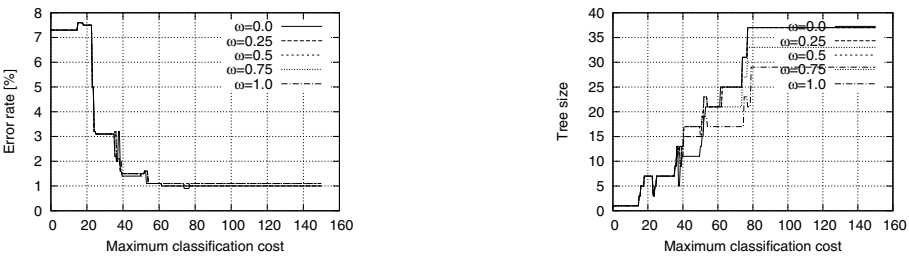
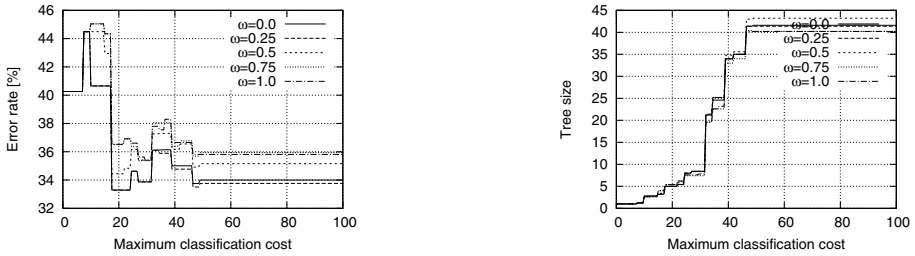
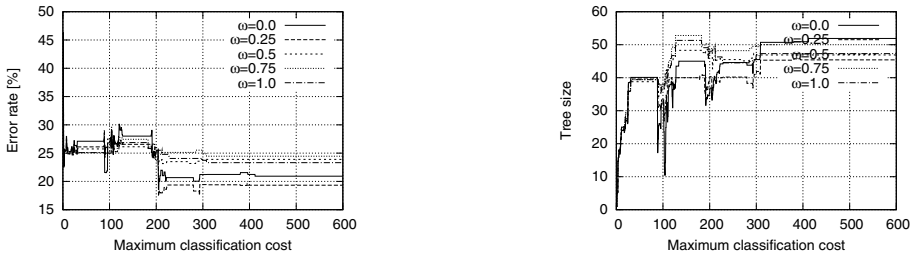


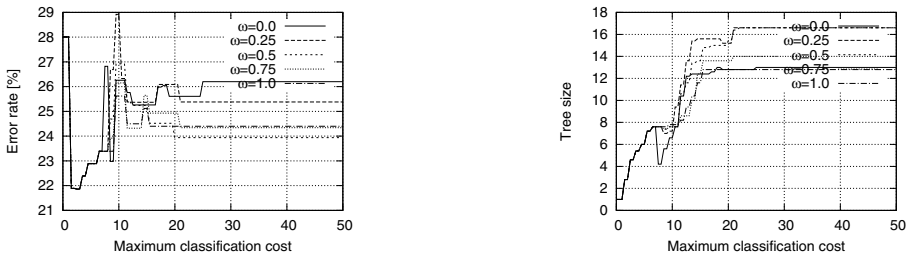
Fig. 2. Classification error and decision tree size versus maximum classification cost for thyroid database



**Fig. 3.** Classification error and decision tree size versus maximum classification cost for liver database



**Fig. 4.** Classification error and decision tree size versus maximum classification cost for heart-disease database



**Fig. 5.** Classification error and decision tree size versus maximum classification cost for hepatitis database

C language. For this purpose we modified the C4.5 algorithm source code. Probabilities of errors of the classifiers were estimated using the 10-cross-validation method. For all experiments five databases from UCI ML Repository [5] were chosen. They concern different fields of medicines. These databases indeed differ from each other in respect of: numbers of examples, amounts and types of attributes, numbers of classes. The details of tested databases are presented in Tab.1.

The results of experimental investigations are presented in Fig. 2-6.

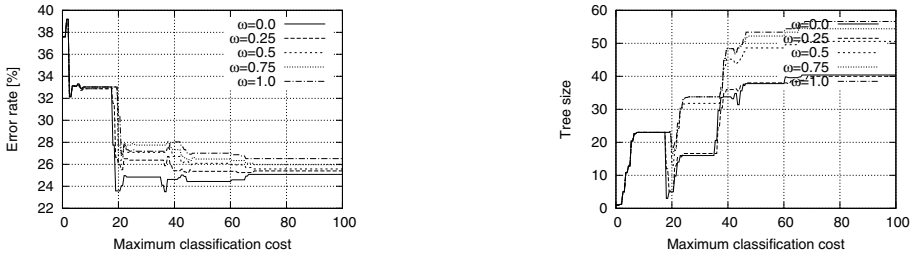


Fig. 6. Classification error and decision tree size versus maximum classification cost for pima-indian-diabetes database

### 5 Experimental Results Evaluation

Generally speaking obtained results did not surprise us and they were easy to predict. We obtained better classifiers in case there are not any cost limits or influence of cost for feature selection is small. But we have to mention interesting phenomenons we have observed in some experiments. For *pima-indian-diabetes* and *pima-indian-diabetes* database we noted that increasing cost limit did not influenced in classification quality improvement. Obtained cheap classifier (for cost limit ca. 20) had good quality and it based on small decision tree structure. The small decision tree size protected obtained classifier from overfitting. Observed effect of decreasing tree size and error rate while increasing cost limit were caused by availability of new, more expensive features, which were better then cheaper ones used previously. At the same time there were not "money" for cheaper attributes used earlier. These cheap features were responsible for building overgrown tree. When cost limit increased, that cheap attributes became available again and they caused tree overgrowth and lower quality of obtained classifier. For another databases like *thyroid*, *liver* and *hepatitis* we observed that there was cost limit for which obtained classifiers were good enough. By increasing the cost limit we obtained classifiers slightly better but tree size grew rapidly, what might have caused generate overtrained classifiers.

### 6 Conclusion

Idea of cost sensitive learning method of decision tree with fixed cost limit was presented in this paper. The properties of proposed concept were established during computer experiments conducted on five medical databases. The obtained results did not surprise but we noted some interesting properties of method under consideration. We hope that presented idea will be useful for constructing real decision systems especially for decision-aided diagnosis in distributed technological environment. For this case the time for decision making is short and cost is interpreted as time necessary to measure any feature value.



## Acknowledgement

This work is supported by The Polish State Committee for Scientific Research under the grant which is realizing in years 2006-2009.

## References

1. Duda R.O., Hart P.E., Stork D.G.(2001) Pattern Classification, Wiley-Interscience.
2. Greiner R., Grove A., Roth D.(1996) Learning active classifiers, in: Proceedings of the 13th International Conference on Machine Learning.
3. Lirov, Y., Yue, O.C.(1991) Automated network troubleshooting knowledge acquisition. *Journal of Applied Intelligence*, 1:121–132.
4. Mitchell T.M.(1997) Machine Learning, McGraw-Hill Comp., Inc, New York.
5. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J. (1998). UCI Repository of machine learning databases [<http://www.ics.uci.edu/mlearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science.
6. Norton, S.W.(1989) Generating better decision trees. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, IJCAI-89, Detroit, Michigan:800–805.
7. Núñez, M.(1988) Economic induction: A case study. Proceedings of the Third European Working Session on Learning, EWSL-88, California: Morgan Kaufmann:139–145.
8. Núñez, M.(1991) The use of background knowledge in decision tree induction. *Machine Learning*, 6:231–250.
9. Quinlan J.R.(1986) Induction on Decision Tree, *Machine Learning*, Vol.1:81-106
10. Quinlan J.R., C4.5(1993) Programms for Machine Learning, Morgan Kaufman, San Mateo, CA.
11. Tan, M., Schlimmer, J.(1989) Cost-sensitive concept learning of sensor use in approach and recognition. Proceedings of the Sixth International Workshop on Machine Learning, ML-89 Ithaca, New York:392-395.
12. Tan, M., Schlimmer, J.(1990) CSL: A cost-sensitive learning system for sensing and grasping objects. IEEE International Conference on Robotics and Automation. Cincinnati, Ohio.
13. Tan, M.(1993) Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning*, 13:7-33.
14. Verdenius, F.(1991) A method for inductive cost optimization. Proceedings of the Fifth European Working Session on Learning, EWSL-91, New York: Springer-Verlag:179–191.
15. Turney P.D.(1995) Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm, *J. Artif. Intell. Res.* 2:369-409.

---

# Fast Adaptive Fourier Transform for Fourier Descriptor Based Contour Classification

Dariusz Puchala and Mykhaylo Yatsymirskyy

Institute of Computer Science Technical University of Lodz  
{dpuchala,jacym}@ics.p.lodz.pl

**Summary.** In this paper the authors presented the results of experiments on evaluation of efficiency of fast adaptive Fourier transform algorithm applied to the tasks of Fourier descriptor based contour classification. The experiments involved popular SQUID boundaries database of sea fish of different species. For object classification a nonparametrised nearest-neighbor classifier was used and distances between feature vectors were expressed in Euclidean metric.

The results received for fast adaptive Fourier transform algorithm indicate significant acceleration of contour descriptors calculation and averagely more than threefold reduction in amount of contour data obtained without loss of classification accuracy.

## 1 Introduction

Automatic classification of 2D objects plays an important role in fields of medical and technical diagnosis, forensic science, topography and satellite image analysis, astronomy, biometrical authorisation systems, etc. [1] - [5]. Numerous comparative studies [6, 7, 8] proved good efficiency of Fourier descriptors in tasks of shape classification. Their basic advantage lies in simplicity of obtaining contour representation invariant to object: translation, rotation and scaling [9]. Moreover due to low pass filtering effect they constitute shape representatives highly immune to contour noise [8]. Practical applications of Fourier descriptors include: shape analysis for mammographic calcifications [1], measurements of acutance of breast tumors [2], automatic classification of teeth in X-ray dental images [3], identification of military ground vehicles [4], classification of topographic shapes [5], etc. Hence the task of improvement of Fourier descriptors is of great importance.

In this paper the results of experiments on one of such improvements are presented. In order to reduce the number of computations involved in process of calculation of Fourier descriptors the authors proposed to utilize the fast adaptive discrete Fourier transform algorithm (AFFT) [10] in the place of fast Fourier transform (FFT). The AFFT algorithm enables automatic selection of such number of samples that are sufficient to calculate spectrum with required accuracy. In the tasks of shape classification it results in selection of the number of contour points that are needed to calculate Fourier descriptors with given

accuracy, where error of descriptors calculation can be estimated in maximum difference (MD) or mean square error metric (MSE). Moreover all calculations within AFFT algorithm are performed in accordance with the structures of fast time decimated algorithms of computational complexity  $O(N \log_2(N))$ . In such manner the number of computations together with the total amount of contour data can be reduced.

The experiments were performed on SQUID boundaries database of sea fish of various species [11], where for contour description the centroid distance signature [9] was selected. Moreover in the process of objects classification a non-parametrised nearest neighbor classifier was employed and distances between feature vectors were expressed in Euclidean metric. The results obtained with AFFT algorithm for several lengths of Fourier descriptors: 8, 16 and 32 coefficients were compared in categories of computational time and the number of contour points with results received for discrete Fourier transform (DFT) and fast radix-2 FFT algorithm [12].

## 2 Theoretical Background

### 2.1 Fourier Descriptors

A continuous and closed 2D contour  $K$  can be described by continuous and periodic (with period  $T$ ) function  $z(t)$  of variable  $t \in [0, T]$ , defined in complex form as  $z(t) = x(t) + iy(t)$ , where  $x$  and  $y$  are spacial coordinates of contour points. In practical solutions for contour description the *centroid distance signature* (see [9]) is frequently selected, which in its continuous form can be written as function

$$r(t) = \sqrt{(x(t) - x_c)^2 + (y(t) - y_c)^2},$$

where  $(x_c, y_c)$  are the coordinates of central point calculated with equations

$$x_c = \frac{1}{T} \int_0^T x(t) dt, \quad y_c = \frac{1}{T} \int_0^T y(t) dt.$$

That signature can be characterized by fast convergence of Fourier descriptors and high accuracy of classification even for short lengths of descriptors composed of only several spectral coefficients (see [8, 9]).

Signature  $r(t)$  is a real-valued and periodic function with period  $T$ . Hence, in accordance with paper [12], it can be reconstructed identically with countable set of Fourier spectral coefficients defined for  $k = 0, 1, \dots, +\infty$  as

$$R(k/T) = \int_0^T r(t) e^{-ikt/T} dt.$$

However for shape classification, as the feature vector describing contour  $K$ , mainly small numbers  $M$  (e.g. 8, 16, 32 coefficients) of low frequency coefficients  $R(k/T)$  are selected, where  $k = 0, 1, \dots, M - 1$ .

In practice of digital signal processing, instead of continuous contour  $K$ , a finite set of its discrete samples is known. Let us assume that signature  $r(t)$  is uniformly sampled with step  $\Delta t = T/N$ , where  $N$  is the number of samples taken at discrete time instants  $t_n = n\Delta t$  for  $n = 0, 1, \dots, N-1$ . Then the centroid distance signature can be described by formula

$$r(n) = \sqrt{(x(n) - x_c)^2 + (y(n) - y_c)^2}, \quad (1)$$

where  $r(n) \equiv r(t_n)$ ,  $x(n) \equiv x(t_n)$  and  $y(n) \equiv y(t_n)$ , while point  $(x_c, y_c)$  is the central point with coordinates defined by equations

$$x_c = \frac{1}{N} \sum_{n=0}^{N-1} x(n), \quad y_c = \frac{1}{N} \sum_{n=0}^{N-1} y(n).$$

In fact as the consequence of rounding  $x$  and  $y$  coordinates to the nearest values resulting from finite horizontal and vertical image resolution the following relation  $r(n) \approx r(t_n)$  takes place. In further considerations an error of coordinates quantization is omitted.

The values of spectral coefficients that constitute Fourier descriptors for discrete signature  $r(n)$  can be calculated with use of discrete Fourier transform

$$R_N(k) = DFT_N \{r(n)\} = \frac{1}{N} \sum_{n=0}^{N-1} r(n) e^{-i \frac{2\pi}{N} kn}, \quad (2)$$

where in general  $R_N(k) \approx R(k/T)$ , i.e. the values of  $R(k/T)$  coefficients can be calculated with some error. That error is due to sampling of  $r(t)$  function and arises as an effect of overlapping of spectrum replicated at integer multiples of sampling frequency  $1/\Delta t$  (i.e. aliasing phenomenon). The value of aliasing error expressed in metric  $B$  will be described symbolically by  $\epsilon_B^R$ .

Fourier descriptors calculated by means of equation (2) must be normalized in order to obtain invariance to such transformations of contour  $K$  as: scaling, rotation and choice of the starting point. Translation invariance ensures construction of  $r(n)$  signature. Invariance to the remaining three transformations can be obtained: by division of modules of coefficients with  $|R(0)|$  factor in case of scaling and for rotation and freedom of the starting point selection by subtracting from arguments of  $R(k)$  coefficients the factor  $\arg(R(1))k$  for  $k = 1, 2, \dots, M-1$ , i.e.

$$\bar{R}(k) = \frac{|R(k)|}{|R(0)|} e^{-i(\arg(R(k)) - \arg(R(1))k)}, \quad (3)$$

where  $\arg(z)$  represents an argument of complex number  $z$ . Thus the normalized feature vector  $\{\bar{R}(k) : k = 1, 2, \dots, M-1\}$  for contour  $K$  can be constructed.

### Fast Radix-2 Fourier Transform Algorithm

The computational complexity of formula (2) is of  $O(MN)$  order. However it can be reduced to  $O(N \log_2(N))$  level with fast algorithm for calculation of discrete

Fourier transform. In this paper the authors utilized popular radix-2 algorithm [12] which can be described by decomposition formula

$$R_N(k) = R_1(k) + R_2(k)e^{-i\frac{2\pi}{N}k}, \quad (4)$$

where  $R_N(k) = DFT_N\{r(n)\}$  is  $N$ -point discrete Fourier transform of  $r(n)$  and  $R_1(k) = DFT_{N/2}\{r(2n)\}$  and  $R_2(k) = DFT_{N/2}\{r(2n+1)\}$  are two  $N/2$ -point DFT transforms of even and odd indexed samples of  $r(n)$  signature respectively. Then by recursive application of equation (4), an  $N$ -point DFT transform can be calculated granted that  $N$  is an integer power of two.

In the considered case spectral coefficients  $R(k)$  are calculated for several values of  $k = 0, 1, \dots, M-1$ , where  $M \ll N$ . It should be noted that FFT algorithm gives acceleration of computations only if  $2^M > N$ .

### Fast Adaptive Fourier Transform Algorithm




Fast adaptive Fourier transform algorithm proposed in paper [10] enables automatic selection of such number  $N_1$  of  $r(t)$  signature samples that are sufficient to calculate  $R_{N_1}(k)$  descriptors with error smaller than given acceptable one  $\epsilon$ . The actual error  $\epsilon_B^R$  of calculation of Fourier descriptors can be estimated in maximum difference or mean square error metrics. In this paper mean square error metric was selected. An estimated value  $\epsilon_{MSE}^O$  of  $\epsilon_{MSE}^R$  error can be expressed in this metric by formula

$$\epsilon_{MSE}^O = \frac{1}{M} \sum_{k=0}^{M-1} \left| R_{N_1}(k) - 2R_{\frac{N_1}{2}}(k) \right|^2 / 9.$$

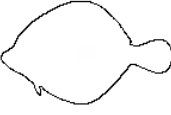


Then the adaptive selection of samples starts with step  $N_1 = 2M$  and consists in recursive doubling of  $N_1$  until inequality  $\epsilon_{MSE}^O \leq \epsilon$  is fulfilled. All calculations within AFFT algorithm are performed in accordance with decomposition formula (4). Hence the computational complexity of AFFT is of  $O(N \log_2(N))$  order. Additional reduction of computations is achieved whenever the estimation  $\epsilon_{MSE}^O$  reaches level smaller than acceptable error  $\epsilon$ . At that time the adaptive calculation of  $R_{N_1}(k)$  for  $k = 0, 1, \dots, M-1$  finishes with  $N_1$  samples, where  $N_1$  number can be significantly smaller than  $N$ .

## 3 Experimental Results

For testing purposes the SQUID [11] contour database of sea fish of different species was exploited. Depending on database object the number  $N$  of contour points oscillated within range of 400 to 1400. From all available objects six classes were selected (144 contours total) that constituted test and training sets. The first one was additionally enriched with objects created by existing objects transformations including: rotations (by 45, 135, 225, 315 angles), horizontal and vertical flips, scaling (scaling factor was randomly taken from 1.5 up to 2.0)

Pipefishes	Rays	Seahorses
		
training set: 3 objects	training set: 7 objects	training set: 2 objects
test set: 96 objects	test set: 480 objects	test set: 60 objects

Soles	Eels	Sharks
		
training set: 9 objects	training set: 3 objects	training set: 4 objects
test set: 432 objects	test set: 108 objects	test set: 180 objects

**Fig. 1.** Object classes with exemplary representatives and numbers of elements constituting test and training sets

and modification of contour points coordinates with additive noise of normal distribution and  $1/12$  variance. In result the test set composed of 1356 objects was obtained. In Picture 1 the assumed classes of objects with their exemplary representatives are presented. For description of contours the centroid distance signature (1) was taken. Feature vectors were constructed on the basis of Fourier descriptors calculated with formula (2) for several numbers  $M = 8, 16$  and  $32$  of coefficients and then normalized by (3) rule.

Because the numbers of contour points for all classified objects were not integer powers of two, then the direct application of fast adaptive AFFT as well as fast radix-2 FFT algorithm for descriptors calculation was not possible. In order to enable that objects contours were interpolated with first order Lagrange polynomials [13] and then sampled with steps producing sample numbers closest and above the original, but being powers of two. In the process of object classification one nearest neighbor 1-NN classifier [9] was exploited and distances between feature vectors were calculated in Euclidean metric.

The obtained results are grouped in three categories related with descriptors calculated for original contours directly with (2) formula (FD), interpolated contours calculated with fast FFT algorithm (IFD) and finally descriptors calculated by means of fast adaptive AFFT algorithm for several values of acceptable errors  $\epsilon$  expressed in MSE metric (AFD). Mean values of descriptor calculation

times and proportional accuracies of classification received for descriptor lengths:  $M = 8, 16$  and  $32$ , and several values of acceptable errors  $\epsilon$  are collected in tables (1) - (4).

By classification accuracy the proportional ratio of properly classified objects number to the total number of objects in the test set is understood

$$\text{Accuracy} = \frac{\text{Number of objects properly classified}}{\text{Number of objects in test set}} \cdot 100\%.$$

The numbers of AFD points describe numbers of contour points needed to obtain the required accuracy of descriptors calculation with given length  $M$ . Those numbers are presented as: minimum values, mean values from all queries and maximum values.

**Table 1.** Mean times of classification for different descriptors lengths  $M$

Descriptor length	Mean time FD [ms]	Mean time IFD [ms]	Mean time AFD [ms]	FD/AFD ratio	IFD/AFD ratio
8	4,87	5,09	1,46	3,33	3,49
16	9,63	4,96	1,58	6,09	3,14
32	18,62	4,83	1,29	14,34	3,74
64	37,4	4,79	1,33	28,12	3,6

**Table 2.** Accuracy of classification in function of  $\epsilon$  for case of  $M = 8$

Acceptable error $\epsilon \cdot 10^3$	Accuracy FD [%]	Accuracy IFD [%]	Accuracy AFD [%]	Number of points AFD		
				min.	mean	max.
1,95313	73,75	73,75	74,41	32	64	256
0,97656	73,75	73,75	74,19	32	128	512
0,48828	73,75	73,75	74,78	64	128	512
0,24414	73,75	73,75	73,6	64	128	512
0,12207	73,75	73,75	73,75	64	128	512
0,06103	73,75	73,75	73,75	64	128	1024

From the analysis of the obtained results it follows that descriptors longer than 16-coefficients gave no accuracy improvement and noise resulting from contour interpolation for IFD and AFD algorithms had no influence on classification results. Moreover for all tested descriptor lengths the value  $\epsilon = 0,06103 \cdot 10^{-3}$  of acceptable error (what corresponds with 14-bit precision) produced for AFFT algorithm the same classification accuracies as obtained with FD and IFD algorithms. However here the mean value of contour samples equaled  $N_1 = 128$  for  $M = 8$  coefficients and  $N_1 = 256$  in other cases. Hence AFFT algorithm yielded essential gain in classification time in comparison with solutions exploiting discrete Fourier transform (almost 14 times faster for  $M = 32$ ) and fast radix-2 algorithm (almost 4 times faster for  $M = 32$ ), what can be viewed in Table (1).

**Table 3.** Accuracy of classification in function of  $\epsilon$  for case of  $M = 16$ 

Acceptable error $\epsilon \cdot 10^3$	Accuracy	Accuracy	Accuracy	Number of points AFD		
	FD [%]	IFD [%]	AFD [%]	min.	mean	max.
1,95313	72,94	72,94	74,48	32	64	256
0,97656	72,94	72,94	74,19	32	128	512
0,48828	72,94	72,94	73,16	64	128	512
0,24414	72,94	72,94	73,23	64	128	512
0,12207	72,94	72,94	72,79	64	256	512
0,06103	72,94	72,94	72,94	128	256	1024

**Table 4.** Accuracy of classification in function of  $\epsilon$  for case of  $M = 32$ 

Acceptable error $\epsilon \cdot 10^3$	Accuracy	Accuracy	Accuracy	Number of points AFD		
	FD [%]	IFD [%]	AFD [%]	min.	mean	max.
1,95313	72,94	72,94	73,89	64	128	256
0,97656	72,94	72,94	73,08	64	128	512
0,48828	72,94	72,94	72,94	64	128	512
0,24414	72,94	72,94	73,01	128	128	512
0,12207	72,94	72,94	72,94	128	256	512
0,06103	72,94	72,94	72,94	128	256	1024

## 4 Summary and Conclusions

On the basis of the obtained results it can be stated that application of AFFT algorithm to calculation of Fourier descriptors gave essential, because several and even several tenfold times acceleration of descriptors calculation process, in comparison with methods exploiting discrete DFT transform, and several times better results as for fast FFT radix-2 algorithm. The computational gain was a consequence of the fact that the adaptive algorithm without loss of classification accuracy required the mean value of 256 contour points, while the total number of points oscillated between 400 to 1400.

Hence the AFFT algorithm can significantly contribute to reduction of disk memory capacities and transmission channel bandwidths required for storage and transmission of classified object data, what in result might enable application of Fourier descriptors in the fields characterized by large volume datasets required for objects description.

## References

1. Liang Shen, Rangayyan R. M., Desautels J. E. L. (1994) Application of Shape Analysis to Mammographic Calcifications. IEEE Trans. on Medical Imaging, vol. 13, no. 2, pp. 263-274, June 1994.
2. Rangayyan R. M., El-Faramawy N. M., Desautels J. E. L., Alim O. A. (1997) Measures of Acutance and Shape for Classification of Breast Tumors. IEEE Trans. on Medical Imaging, vol. 16, no. 6, pp. 799-810, December 1997.



3. Mohammad Hossein Mahoor, Mohamed Abdel-Mottaleb (2004) Automatic Classification of Teeth in Bitewing Dental Images. ICIP International Conference On Image Processing, pp. 3475-3478, October 2004.
4. Sun-Gu Sun, Junsung Park, Yong Woon Park (2003) Identification of Military Ground Vehicles by Feature Information Fusion in FLIR Images. Proc. 3rd International Symposium on Image and Signal Processing and Analysis, pp. 871-876, 2003.
5. Keyes L., Winstanley A. (1999) Fourier Descriptors as a General Classification Tool for Topographic Shapes. Proc. Irish Machine Vision and Image Processing Conference, pp. 193-203, 1999.
6. Ezer N., Anarim E., Sankur B. (1994) A Comparative Study of Moment Invariants and Fourier Descriptors in Planar Shape Recognition. Proc. 7th Mediterranean Electrotechnical Conference, April 1994.
7. Kauppinen H., Seppänen T., Pietikäinen M. (1995) An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification. Trans. on Pattern Analysis and Machine Intelligence, vol. 17, no. 2, pp. 201 - 207, February 1995.
8. Osowski S., Do Dinh Nghia (2002) Fourier and Wavelet Descriptors for Shape Recognition Using Neural Networks - A Comparative Study. Pattern Recognition, vol. 35, no. 9, September 2002.
9. Dengsheng Zhang, Guojun Lu (2002) A Comparative Study of Fourier Descriptors for Shape Recognition and Retrieval. Proc. 5th Asian Conference On Computer Vision, January 2002.
10. Puchala D., Yatsymirskyy M. (2006) Fast Adaptive Algorithm for Fourier Transform. Proc. of International Conference on Signals and Electronic Systems, pp. 183-185, d, 2006.
11. <http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html>
12. Cooley J. W., Lewis P. A., Welch P. D. (1967) Application of the Fast Fourier Transform to Computation of Fourier Integrals, Fourier Series, and Convolution Integrals. IEEE Trans. on Audio and Electroacoustics, vol. AU-15, no. 2, pp. 79-84, June 1967.
13. Allen III M. B., Isaacson E. L. (1997) Numerical Analysis for Applied Science. John Wiley & Sons, Inc. 1997.

---

# Separable Decomposition of Graph Using $\alpha$ -cliques

Henryk Potrzebowski<sup>1</sup>, Jarosław Stańczak<sup>2</sup>, and Krzysztof Sep<sup>3</sup>

<sup>1</sup> System Research Institute PAS, ul. Newelska 6, 01-447 Warsaw  
henryk.potrzebowski@ibspan.waw.pl

<sup>2</sup> System Research Institute PAS, ul. Newelska 6, 01-447 Warsaw  
jaroslaw.stanczak@ibspan.waw.pl

<sup>3</sup> System Research Institute PAS, ul. Newelska 6, 01-447 Warsaw  
sep@ibspan.waw.pl

**Summary.** The article deals with an evolutionary based method to decompose graph into strongly connected structures, we called  $\alpha$ -cliques. The  $\alpha$ -clique is a generalization of a clique concept with the introduction of parameter  $\alpha$ . Using this parameter it is possible to control the degree (or strength) of connections among vertices (nodes) of this sub-graph structure. The evolutionary approach is proposed as a method that enables to find separate  $\alpha$ -cliques that cover the set of graph vertices.

## 1 Introduction

In this paper we propose a new method that allows obtaining a separable decomposition of graph into subsets [2, 3, 4, 9] -  $\alpha$ -cliques. Vertices in each subset are relatively stronger connected with each other than with vertices belonging to different subsets. Sometimes it is difficult or even impossible to divide a graph into separable cliques, or obtained cliques are very small. Introduction of the parameter  $\alpha$ , which shows the minimal percent of sub-graphs nodes that must be connected with every vertex of this sub-graph, allows to control the strength of connections among vertices in  $\alpha$ -clique and allows to obtain sub-graphs of size which depends on the accepted value of parameter  $\alpha$ . Thus using  $\alpha$  we may tune the size of computed sub-graphs to requirements of solved problem. For  $\alpha=1$  an  $\alpha$ -clique becomes a traditional clique – each node must be connected with all remaining nodes in sub-graph (we assume that each node is bounded with himself). As it can be seen the idea of  $\alpha$ -clique is an expansion of well known clique concept and can be applied in many domains, including designing of computer, teletransmission or similar networks, electronic circuits, air connections, transportation, clustering etc [11, 16, 17]. Unfortunately, no specialized or traditional algorithms to find  $\alpha$ -cliques have been prepared so far. Thus evolutionary approach, described in this article, seems to be a good tool to derive  $\alpha$ -cliques. The evolutionary algorithm provides an efficient tool that enables to obtain solutions of this rather difficult computational problem [2, 3, 7, 10, 20].

## 2 Basic Concepts

A **graph** is a pair  $G = (V, E)$ , where  $V$  is a non-empty set of vertices and  $E$  is a set of edges. Each edge is a pair of vertices  $(v_1, v_2)$  that  $v_1 \neq v_2$  [22].

Two vertices in graph  $G = (V, E)$  we call **incident** if  $v_i, v_j \in V$  then  $\{v_i, v_j\} \in E$ .

**Sub-graph** of graph  $G = (V, E)$  is graph  $G' = (V', E')$ , where  $V' \subseteq V$  and  $E' \subseteq E$  that for all  $e \in E$  and  $e = v_1, v_2$  if  $v_1, v_2 \in V'$  then  $e \in E'$  [22].

**Path** in graph  $G = (V, E)$  from vertex  $s$  to vertex  $t$  is a sequence of vertices  $\{v_1, \dots, v_n\}$  that:  $\{s, v_1\} \in E, \{v_i, v_{i+1}\} \in E$  for  $n = 1, 2, \dots, n - 1, \{v_n, t\} \in E$  [22].

Graph  $G = (V, E)$  is a **connected graph**, if for each pair of vertices there is a path between them [22].

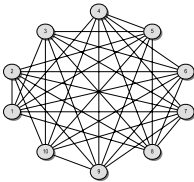
A **clique** (a complete sub-graph)  $Q = (V_q, E_q)$  in graph  $G = (V, E)$  is a graph that  $V_q \subseteq V$  and  $E_q \subseteq E$  and each pair of vertices  $v_1, v_2 \in V_q$  fulfills the condition  $v_1, v_2 \in E_q$ . [8].

**The maximum clique**  $Q_M = (V_q, E_q)$  in graph  $G = (V, E)$  is a clique, where there is no vertex  $v \in V$  and  $v \notin V_q$  that  $Q' = (V', E')$  is a clique, where  $V' = V \cup \{v\}$  and  $E' \subseteq E$  and each pair  $v_1, v_2 \in V'$  of vertices fulfills the condition  $\{v_1, v_2\} \in E'$  [6, 8, 10, 18].

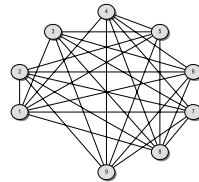
An  $\alpha$ -clique

Let  $A = (V, E)$  is a sub-graph of graph  $G = (V, E), V \subseteq V, E \subseteq E, k = Card(V), k_i$  is a number of vertices  $v_j \in V$  that  $\{v_i, v_j\} \in E$

For  $k = 1$  or  $k > 1$ , for all vertices  $v_i \in V$  fulfill the condition  $\alpha \leq \frac{k_i + 1}{k}$ , the graph  $A$  is an  $\alpha$ -clique in graph  $G$ , where  $\alpha \in (0, 1]$ .



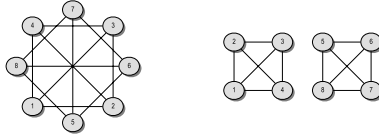
**Fig. 1.** An example of  $\alpha$ -clique for



**Fig. 2.** A sub-graph of graph from Fig 3. which is not  $\alpha$ -clique for

As it can be seen in Fig. 1 and 2 a sub-graph of  $\alpha$ -clique ( $\alpha = 0.8$ ) is not an  $\alpha$ -clique ( $\alpha = 0.8$ ), thus the property of being  $\alpha$ -clique with some value of parameter  $\alpha$  may not be preserved by its sub-graphs.

Let  $\alpha$ -clique  $A = (V, E)$  is a graph with, thus for all of vertices  $v_i$  belong to  $\alpha$ -clique  $k_i + 1 > 0.5k$ . From set theory, for all two vertices, sets of vertices incident with each of them have a non-empty intersection, so the  $\alpha$ -clique with  $\alpha > 0.5$  constitutes a connected graph. If  $\alpha = 0.5$  the obtained sub-graph may be disconnected - an example of such situation is shown in Fig. 3.



**Fig. 3.** An example of not connected graph for  $\alpha = 0.5$

### 3 The Evolutionary Approach to Find $\alpha$ -clique

Standard evolutionary algorithm (EA) works in the manner as it is shown in the Algorithm 1, but this simple scheme requires many problem specific improvements to work efficiently. The adjustment of the genetic algorithm to the solved problem requires a proper encoding of solutions, an invention of specialized genetic operators for the problem, accepted data structure and a fitness function to be optimized by the algorithm.

1. Random initialization of the population of solutions.
2. Reproduction and modification of solutions using genetic operators.
3. Valuation of the obtained solutions.
4. Selection of individuals for the next generation.
5. If a stop condition is not satisfied, go to 2.

**Algorithm 1.** The evolutionary algorithm

#### 3.1 Individual Representation

The whole information about the problem is stored in an array of data that describes all data connections. This array can be binary (a matrix of incidence of undirected graph: 0 – no connection, 1– presence of connection) or non-negative (undirected graph) real-valued and in this case the stored value denotes the strength of the connection. Members of the population contain their own solutions of the problem as a dynamic table of derived  $\alpha$ -cliques (number of them may change during computations). Each element of this table ( $\alpha$ -clique) has a list of nodes attached to this  $\alpha$ -clique and each node is considered only once in one solution (population member). Unattached nodes are also included, they constitute small, one-element  $\alpha$ -cliques (one node is also  $\alpha$ -clique with  $\alpha=1$ ). Thus each solution contains all nodes from a graph described by incidence matrix. But the solution with many small  $\alpha$ -cliques is rather not profitable, and it is the role of evolutionary algorithm to find bigger ones. Beside it, the member of the population contains several more data including: a vector of real numbers, which describe its knowledge about genetic operators and a number of the operator chosen to modify the solution in the current iteration. More details about genetic operators and the method of valuation them will be given later in this chapter.

### 3.2 Fitness Function

The problem's quality function is closely connected with the fitness function, which evaluates the members of the population. In the solved problem several quality functions may be considered, depending on input data (binary, integer or real) or what set of  $\alpha$ -cliques one want to obtain (equal size or maximal size etc.). The fitness function does not have to possess any punishment part for an  $\alpha$ -clique constraint violation, because forbidden solutions are not produced by initializing function or genetic operators. Thus all population members contain only valid  $\alpha$ -cliques with their local values of  $\alpha$  not less than the global value imposed on the solved problem. For computer simulations we used fitness function as follows:

$$\max Q = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{l_i} \sum_{k=1}^j D[t_{ik}, t_{ij}] \quad (1)$$

where:

$n$  numbers of  $\alpha$ -cliques in the valued solution;

$l_i$  – number of nodes in the  $i$ -th  $\alpha$ -clique;

$D$  – the data array (incidence matrix);

$t_{ij}, t_{ik}$  – nodes of the  $i$ -th  $\alpha$ -clique.

The fitness function (1) promotes  $\alpha$ -cliques of medium size (if possible - almost equal) and was used to solve testing problems, but it is possible to propose different fitness functions [16] with various properties of sizes of generated  $\alpha$ -cliques.

### 3.3 Specialized Operators

The described data structure requires specialized genetic operators, which modify the population of solutions. Each operator is designed in such a manner that preserves the property of being  $\alpha$ -clique for the modified parts of solutions. If modified solution violates the limitation of being  $\alpha$ -clique, the operation is cancelled and no modification of solution is performed. This method is more difficult for the evolutionary algorithm to find satisfying solutions, due to possible bigger problems with local maximums, than method with penalty function, but it gives the certainty that computed solutions are allowed.

Designed genetic operators are:

1. mutation – an exchange of randomly chosen nodes in different  $\alpha$ -cliques;
2. movement of randomly chosen node to a different  $\alpha$ -clique;
3. “intelligent” movement – performed only if this modification gives better value of fitness function;
4. concatenation – tries to concatenate (mainly small)  $\alpha$ -cliques;
5. also multiple versions of operators are applied.

### 3.4 Evolutionary Algorithm Used to Solve the Problem

Using of specialized genetic operators requires applying some method of sampling them in all iterations of the algorithm. In the used approach [21] it is assumed that an operator that generates good results should have bigger probability and more frequently effect the population. But it is very likely that the operator, that is proper for one individual, gives worse effects for another, for instance because of its location in the domain of possible solutions. Thus every individual may have its own preferences. Every individual has a vector of floating point numbers, beside encoded solution. Each number corresponds to one genetic operation. It is a measure of quality of the genetic operator (a quality factor). The higher the factor is, the higher is the probability of the operator. The ranking of qualities becomes a base to compute the probabilities of appearance and execution of genetic operators. Simply normalization of the vector of quality coefficients turns it into a vector of operators' execution probability. The set of probabilities is also a base of experience of every individual and according to it, an operator is chosen in each epoch of the algorithm. Due to the gathered experience one can maximize chances of its offspring to survive.

### 3.5 Computer Simulations

First we wanted to know how the decomposition of graph into  $\alpha$ -cliques depends on the accepted value of parameter  $\alpha$ . We used as testing examples data from BHOSLIB: Benchmarks with Hidden Optimum Solutions for Graph Problems (Maximum Clique, Maximum Independent Set, Minimum Vertex Cover and Vertex Coloring) – Hiding Exact Solutions in Random Graphs <http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/graph-benchmarks.htm>. The first task was rather a large graph with 4000 vertices and 7 425 226 edges (test frb100-40.clq.gz), the second problem was a little smaller graph with 4000 vertices and 572 774 edges (test frb100-40mis.gz). As it can be seen on Tab. 1 and 2, the value of  $\alpha$  has a decisive impact on the number of generated sub-graphs and their average sizes. Of course our method cannot exceed limitations of tested graphs. The graph from the first example is almost fully connected, thus only using higher than 0.8 values of  $\alpha$  it is possible to control parameters of derived  $\alpha$ -cliques. Similarly the graph from the second example is very

**Table 1.** Obtained results for several values of  $\alpha$  and a graph with 4000 vertices and 7 425 226 edges

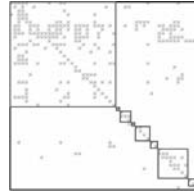
$\alpha$	Min $\alpha$ -clique	Max $\alpha$ -clique	No of $\alpha$ -cliques	Av. size of $\alpha$ -clique
0,80	4000	4000	1	4000
0,90	41	607	15	267
0,95	1	277	39	103
0,97	8	131	56	71
0,99	5	74	78	51
1,00	10	73	76	53

**Table 2.** Obtained results for several values of  $\alpha$  and a graph with 4000 vertices and 572 774 edges

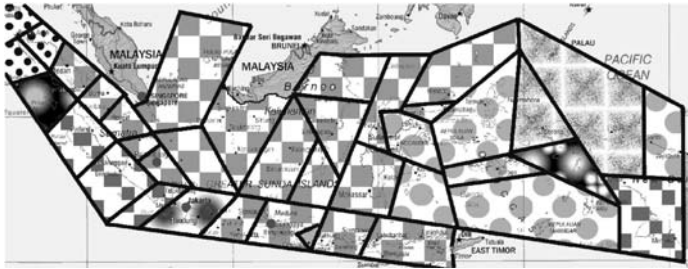
$\alpha$	Min $\alpha$ -clique	Max $\alpha$ -clique	No of $\alpha$ -cliques	Av. size of $\alpha$ -clique
0,51	40	119	50	80
0,60	6	111	67	60
0,70	2	80	94	43
0,80	40	40	100	40
0,90	40	40	100	40
0,99	40	40	100	40
1,00	40	40	100	40



**Fig. 4.** The source data



**Fig. 5.** The solution obtained with  $\alpha = 0.7$  which is not  $\alpha$ -clique for



**Fig. 6.** The map of trade connections among Indonesian regions

weakly connected, so only smaller than 0.8 values of  $\alpha$  can affect properties of obtained  $\alpha$ -cliques. It is also interesting to mention about times of performed computations.

The example deals with an import-export analysis where rows and columns represent a set of 50 regions of the Indonesian islands. In the source data  $a_{ij} = 1$  if in 1971 a quantity of at least 50 tons of rice was transported from region  $i$  to region  $j$ , and  $a_{ij} = 0$  otherwise. The aim of the analysis is to find trade connections among different regions of Indonesia. Obtained results are illustrated on Fig. 4 - data clustering and its interpretation on the maps of Indonesia (Fig. 6).

This or similar problems were also solved using different methods: greedy [11], 2-opt [11], 3-opt [11, 19] or BEA – Bounded Energy Algorithm [11, 13] or even EA [1, 12, 14, 15, 19, 23]. The mentioned algorithms were used only as “pre-processing” methods to optimize the source data matrix (data preprocessing to obtain more condensed groups of “1” - existing connections) and then to extract clusters from preprocessed data. Our approach is simple and allows obtaining good results using one method with a possibility to control the results using different values of parameter  $\alpha$ .

### 3.6 Conclusions

It's well known for problems with large-scale complexity that there are no effective algorithms to solve them, but possibility to construct approximation ones is still open. Generally  $\alpha$ -clique problem considered in the paper probably can be solved using some greedy-like method, but at the moment we developed the evolutionary method that efficiently solves that problem.

Results of series of conducted experiments are rather optimistic, parameter  $\alpha$  introduced to traditional clique notion makes it a flexible tool that enables solving of graph decomposition problems with the ability of tuning the obtained sub-graphs parameters using different values of  $\alpha$ .

## References

1. Altus S S, Kroo I M, Gage P J, (1996) Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems. *Journal of Mechanical Design*, Vol. 118, No. 4.
2. Bagirov A M, Yearwood J. (2006) A new non-smooth optimization algorithm for minimum sum-of-squares clustering problems, Centre for Informatics and Applied Optimization, School of Information Technology and Mathematical Sciences, University of Ballarat, P.O. Box 663, Vic. 3353, Australia.
3. Benson S J, Ye Y, (2000) Approximating maximum stable set and minimum graph coloring problems with the positive semi-definite relaxation, *Computational complexity: The problem of approximation*, Kluwer Academic Publishers.
4. Browning T R, (2001) Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions, *IEEE Transactions on Engineering Management*, Vol. 48, No 3.
5. Cichosz P, (2000) *Systemy uczące się* (in Polish), WNT, Warszawa.
6. Hansen P, Mladenovic N, Urosevic D, (2004) Variable neighborhood search for the maximum clique The Fourth International Colloquium on Graphs and Optimisation (GO-IV) GO-IV International Colloquium on Graphs and Optimisation No4, Loèche-les-Bains, SUISSE (20/08/2000), vol. 145, no 1 (28 ref.), pp. 117-125.
7. Hassin R, Khuller S, (1986)  $\alpha$ -Approximation, *Computational complexity: The problem of approximation*, New York.
8. Jukna S, (2001) *Extremal Combinatorics*, Springer-Verlag Berlin Heidelberg.
9. Kloks T, Kratsch D, (1998) Listing all minimal separators of a graph, *SIAM J. COMPUT.* Vol. 27, No. 3, pp. 605-613.
10. Kumlander D, (2007) An Approach for the Maximum Clique Finding Problem Test Tool Software Engineering, Software Engineering, SE 2007, Innsbruck, Austria



11. Lenstra J K, (1997) Sequencing by Enumerative Methods, Mathematisch Centrum, Amsterdam.
12. McCulley C, Bloebaum C A, (1996) Genetic tool for optimal design sequencing in complex engineering systems, Structural Optimization, Vol. 12, No. 2-3.
13. Owsiański J W, (1990) On a new naturally indexed quick clustering method with a global objective function, Applied Stochastic Models and Data Analysis, Vol. 6.
14. Potrzebowski H. Stańczak J, Sęp K, (2004) Evolutionary method in grouping of units with argument reduction, Proceedings of the 15th International Conference on Systems Science, ed. Z. Bubnicki, A. Grzech, vol. III, pp. 29-36.
15. Potrzebowski H. Stańczak J, Sęp K, (2005) Evolutionary method in grouping of units Proceedings of the 4th International Conference on Recognition Systems CORESŠ05, Springer-Verlag, Berlin-Heidelberg.
16. Potrzebowski H. Stańczak J, Sęp K, (2006) Evolutionary Algorithm to Find Graph Covering Subsets Using  $\alpha$ -Cliques, Praca zbiorowa pod red. J. Arabasa: Evolutionary Computation and Global Optimization, 351-358. Prace naukowe Politechniki Warszawskiej.
17. Potrzebowski H. Stańczak J, Sęp K, (2006) Heurystyczne i ewolucyjne metody znajdowania pokrycia grafu, korzystające z pojęcia alfa-kliki i innych ograniczeń (in Polish), Badania operacyjne i systemowe 2006. Metody i techniki, Akademicka Oficyna Wydawnicza EXIT, Warszawa.
18. Protasi M, (2001) Reactive local search for the maximum clique problem, Algoritmica vol. 29, no 4. pp. 610-637.
19. Rogers J L, (1997) Reducing Design Cycle Time and Cost Thorough Process Re-sequencing, International Conference on Engineering Design ICED 1997, Tampere, Finland.
20. Sysło M M, Deo N, Kowalik J S, (1983) Algorithms of discrete optimization, Prentice-Hall.
21. Stańczak J, (2003) Biologically inspired methods for control of evolutionary algorithms, Control and Cybernetics, 32(2), pp. 411-433.
22. Wilson R J, (1996) Introduction to graph theory Addison Wesley Longman.
23. Yu T L, Goldberg D E, Yassine A, Yassine C A, (2003) Genetic algorithm design inspired by organizational theory, Genetic and Evolutionary Computation Conference Chicago, Illinois, USA, Publ. Springer-Verlag, Heidelberg, Lecture Notes in Computer Science, Vol. 2724/2003.

---

# A Feature Selection Approach in Problems with a Great Number of Features

Pawel Kosla

Technical University of Lodz, Computer Engineering Department,  
Al. Politechniki 11, 90-924 Lodz, Poland  
pkosla@kis.p.lodz.pl

**Summary.** A misclassification rate is most often used as a feature selection criterion. However, in the cases, when the numerical force of the training set is not sufficiently large in relation to the number of features, the risk of choosing the noisy features is very high. It produces difference between error estimations derived on the basis of the training and testing sets, so the error rate estimation can not be sufficiently confident. Feature preselection based on analysis of dependences between features is recommended in such types of tasks. An advantage of this approach is shown in the paper. As a feature selection criterion the Pearson chi-square statistics has been used.

## 1 Introduction

Many pattern recognition applications have very high requirements concerning the speed and classification quality. The quality depends on the analyzed features. There can be some attributes with small or destructive influence on classifier decision. The speed is also related to feature space size. Therefore, it is recommended to apply the feature selection, which may improve the classification process.

The feature selection, based on a misclassification rate, selects the feature subset, which promises the best classification quality. For reference sets with small number of objects the probability of classification is mostly estimated by the *cross-validation* or *leave one out method* [1]. It is caused by impossibility of extracting validation set from reference one.

Due to this aspect and large number of features the algorithm might find features, which strongly minimize a classification error. Unfortunately, minimization of misclassification rate concerns only training set. In other words, the feature subset adapts to training set. Additionally, in this type of tasks feature selection method strongly reduces dimensionality.

This phenomenon can be discovered by comparing the error rate estimations derived on the basis of the training and testing sets. Large difference between these errors, especially when the first is lower than the second, is very disadvantageous. In the real application only training error can be estimated and this

misclassification rate is expected for right classification. Thus, the training and testing error rates should be occurred on the similar level.

Feature selection, which omits estimation of misclassification rate, will diminish the described problem. Feature selection method should not use information about object membership. Therefore, procedure based on analysis of dependences between features is recommended for data sets with large number of features in relation to amount of objects.

A feature, which is very strongly correlated with other features, has minimal influence on classifier decision. Thus, there is no necessity of taking into consideration this feature in the classification process. An approach proposed in the paper is based on this idea. For examination of interdependence between attributes the chi-square statistics is used. After the preselection process the modified *forward feature selection* algorithm is applied.

## 2 Assumptions and Theoretical Background

The analyzed problem may appear for different types of classifier. The  $k$ -NN classifier is chosen for consideration in the presented paper, because it is quite simple and offers very good performance [1]. The proposed approach can be applied for other classifiers, because the main part (the preselection process) does not change itself.

The most often used feature reduction procedure, i.e. *forward feature selection*, is taken to compare the use an error rate as a feature selection criterion with the criterion based on the chi-square statistics. The Pearson's chi-square statistics is applied for analysis of dependences between features. The main advantage of this test is the speed of process. Moreover, it enables to analyze qualitative features. Furthermore, the modified forward selection algorithm is applied to select final feature subset after the preselection process.

### *Forward feature selection*

This procedure is based on sequential adding single features starting with the feature promising the lowest error rate. So, in the first phase the current selected feature set contains only one feature. In each next step, the feature set is increased by one feature, chosen from the remaining set of attributes. The feature, that added to the currently chosen features offers the smallest misclassification rate, is selected. Thus, the current selected feature set contains sequentially one, two, three, etc. attributes. Finally, the feature set consists of all, let say,  $n$  features available in the training set. The feature set chosen out of these  $n$  currently selected feature set, associated with the lowest error rate forms the final selected feature combination.

### *Pearson chi-square statistics*

This is the most common statistics used for verification of the relationship between categorical variables.

The chi-square test is based on contingency table, which shows frequencies for particular combinations of values of two discrete variables. For variables  $X$  and  $Y$  with following categories:  $X_1, X_2, \dots, X_n$  and  $Y_1, Y_2, \dots, Y_m$  the each element  $N_{ij}$  of the contingency table denotes the number of observations, when simultaneously variable  $Y$  has value  $Y_i$  and  $X$  - variant  $X_j$ . The Pearson [2] statistics for this type of the table is defined as follows:

$$\chi^2 = \sum_{i=1}^m \sum_{j=1}^n \frac{(N_{ij} - E_{ij})^2}{E_{ij}}, \tag{1}$$

where expected values  $E_{ij}$  are estimated by formula (2):

$$E_{ij} = \frac{\sum_{k=1}^m N_{kj} \cdot \sum_{l=1}^n N_{il}}{\sum_{k=1}^m \sum_{l=1}^n N_{kl}}. \tag{2}$$

There are some methods of objective assessment of variables interdependence [3, 4]. The Pearson test utilizes table of critical values of chi-square distributions.

### 3 Description of the Proposed Algorithm

Most often features are not discrete and the digitization of continuous attributes is necessary to apply the chi-square statistics. The range of feature values may be divided into some intervals. The number of divisions is assessed by well known formulas (formula 3 and 4), which are used to construct histograms.

$$k = 1 + 3 \cdot \log(m), \tag{3}$$

$$k \leq 5 \cdot \log(m), \tag{4}$$

where:

$k$  – number of divisions

$m$  – number of observations.

The analysis of the relation between two continuous features can be in such a way replaced by the analysis of dependence between two quantitative variables. Moreover, in each case, number of observations and degrees of freedom are constant and for this reason the chi-square statistics can be directly used as a criterion of feature dependence strength.

For the further analysis the determination of the matrix containing values of Pearson statistics for every pair of features may be helpful.

Proposed approach consists of two phases:

#### *Preliminary arrangement of features*

The preselection phase consists in arranging the attributes without using the information concerned object class-membership. This process is based only on

a measurement of dependence between features. At the beginning two weakest correlated attributes are chosen. The chi-square statistics for these features is minimal. After this step, the feature subset containing two ordered attributes is obtained - the feature with greater standard deviation is preceded by the feature with the lower standard deviation. This subset is temporarily called as "ordered feature set"  $F'_l$  with  $l = 2$  attributes. The other features are collected in the "remaining feature set"  $F_{n-l}$ .

Then, the features from remaining set are sequentially transferred to the ordered one. At the  $l = l + 1$  position the attribute  $f(j) \in F_{n-l}$ , which is weakest related to ordered feature set  $F'_l$ , is placed. Examination which attribute  $f(j)$  is the weakest related with the set  $F'_l$  consists in finding maximum value of chi-square statistics from each pair of features  $f(j) \in F_{n-l}$  and  $f'(i) \in F'_l$ . The smallest value from among of these maxima corresponds to the feature added to the set  $F'_l$  and placed on  $l$ -th position. This procedure is repeated until all features from the set  $F_{n-l}$  will be transferred into the set  $F'_l$ .

#### *Modified forward feature selection*

The second part of the proposed technique is selecting features from the ordered feature set defined previously. This method is the simplified version of the *forward feature selection*. The one, two, ...,  $n$  first features from ordered set are sequentially taken into consideration and misclassification rates are estimated by *leave one out* method. The feature collection promising the lowest error rate is finally selected feature combination.

## 4 Description of the Experiment

### 4.1 Applied Reference Sets

To test the proposed approach two real data sets were used: Ferrites and Satimage. The Ferrites data concerns quality control of ferrite cores [5]. Each sample described by 30 features can be classified to one of 8 classes. The origin data set includes over 5900 patterns. However, the class numerical force was not equal. To obtain a set with an equal number of objects in every class, the data has been reduced to 1440 patterns (8 classes, 180 objects each). The Satimage data concerns to a photographic image recognition problem. It is used to recognize six types of a soil. Data set contains 6435 satellite photographs described by 36 image properties [6].

The training set is formed from randomly chosen patterns from the original data set. Remaining samples create the testing set. According to the initial assumptions, the number of objects in training part should be small in relation to feature space size. Two cases are considered, when number of objects is a nearly equal and when it is about twice as large as the number of features. For Ferrites the training sets with 32 (4 patterns in 8 classes) and 56 (7x8) objects were chosen for examinations. The 36 and 72 objects were chosen for construction the Satimage training sets.

## 4.2 Experimental Design

The four experimental schemes, which in different degree use the object class-membership knowledge, are considered in the presented paper:

1. The traditional forward feature selection.
2. The misclassification rate, estimated by *leave one out* procedure, is evaluated for each single attribute. The ordered feature set is arranged according to the growing values of the error rates.
3. The feature arrangement is based on chi-square statistics. For each feature its dependence on class number is analyzed.
4. The feature arrangement is in accordance with procedures presented in the section 3.

In the schemes 2, 3 and 4, the reviewed feature combinations contain the first one, two, three and so on, features and the combination that offers the smallest error rate, found by the *leave one out* method, is selected (see paragraph *Modified forward feature selection*).

The probabilities of classification in the training and testing phases were analyzed. The operations: dividing data in training and testing sets, feature selection and error rate estimation were repeated 31 times. The experiment was carried out for all four described above feature selection arrangements and for all data sets.

## 5 Results and Discussion

The parameters of error distributions like mean value and standard deviation are presented in tables below (table 1, 2). The degrees of feature reduction for the compared methods are collected in the table 3. When only misclassification rates derived by testing set (called testing error rate) are computed, one can conclude that all presented methods do not differ. The difference can be seen in differences of these error rates computed on the basis of the training and the testing sets.

The method 4 provides lowest reduction (57% on average), but it does not mean that the selection algorithm works weakly. Construction of *forward feature selection* method causes the strong elimination of features – almost 70% reduction. This is an effect (mentioned in introduction) of adapting selected attributes to training set. As we can see in the table 1, the methods, which use more information about classes (i.e. in order 1, 2, 3, 4), promise lower training error rates. In these cases this is disadvantageous outcome, because it causes large disproportion between testing and training errors. The differences of these errors are presented on the figure 1. Among three most similar procedures (except the method 1), the last method based on the relation between features analysis assures the best result, the error differences are remarkably smaller. The problem intensifies for smaller data sets. Methods 2 and 3 achieve comparable outcome because utilize the class-membership information, but in different manners.

**Table 1.** The average values and dispersions of error rates estimated on training sets: method 1 - forward feature selection, method 2 - misclassification error rate criterion, method 3 - chi-square statistic criterion for estimating relation between features and classes, method 4 - proposed approach based on feature interdependence analysis

Data set	Training error rate [%]			
	Method 1	Method 2	Method 3	Method 4
Ferrites56	14, 9 ± 5, 2	20, 3 ± 5, 6	22, 2 ± 6, 7	21, 9 ± 5, 6
Ferrites32	17, 6 ± 7, 7	26, 5 ± 7, 8	27, 9 ± 8, 9	30, 0 ± 8, 2
Satimage72	9, 6 ± 3, 0	13, 5 ± 3, 8	13, 6 ± 3, 5	15, 4 ± 4, 0
Satimage36	9, 2 ± 4, 8	14, 7 ± 6, 8	13, 3 ± 6, 3	18, 1 ± 6, 8

**Table 2.** The average values and dispersions of error rates estimated on testing sets

Data set	Testing error rate [%]			
	Method 1	Method 2	Method 3	Method 4
Ferrites56	27, 8 ± 3, 3	27, 3 ± 3, 0	29, 9 ± 3, 8	28, 6 ± 3, 6
Ferrites32	33, 1 ± 4, 7	35, 4 ± 6, 8	34, 2 ± 4, 1	32, 6 ± 4, 3
Satimage72	20, 3 ± 2, 2	20, 0 ± 2, 4	19, 9 ± 2, 3	20, 6 ± 2, 2
Satimage36	24, 6 ± 3, 7	24, 6 ± 4, 6	22, 7 ± 3, 6	24, 5 ± 3, 9

**Table 3.** Degree of feature reduction by different methods

Data set	Reduced feature set size			
	Method 1	Method 2	Method 3	Method 4
Ferrites56 (30 features)	8	13	9	13
Ferrites32 (30 features)	6	11	9	13
Satimage72 (36 features)	8	13	14	17
Satimage36 (36 features)	6	10	10	13

It is very important to receive the same range of error rates estimated on the training and testing sets, because in the real tasks only the first one can be estimated. In the problems with the great number of features extraction of samples to the testing set is not recommended. It would cause even worse relation between the training set size and the number of features. Nevertheless, proposed approach (method 4) still produces the differences between the error rates calculated for training and testing sets. It is caused by applying the procedures based on the misclassification rate as a criterion after preliminary feature arrangement.

Apart from quality, the proposed method works faster than the others using *leave one out* algorithm. For the Ferrites56 set the defined above approaches as the methods 1, 2, 3 and 4 took 6.4, 0.8, 0.5 and 0.5 seconds respectively.

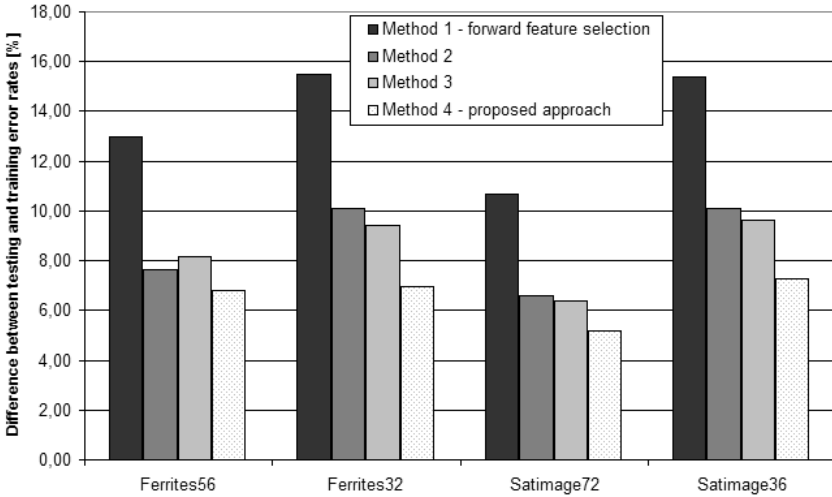


Fig. 1. Analysis of disproportion between testing and training error rates

### More Samples in Relation to the Number of the Features

For the preliminary feature arrangement the samples from the training set as well as the ones to be classified may be used. It is not forbidden because in such a case no advantage of object class-membership is taken. To show the advantage of this approach the objects collected in testing set can be used. This modification improves the classification parameters for proposed feature selection procedure (see table 4).

Table 4. The influence of changing amount of samples used for preliminary feature arrangement

Data set	Training set		All patterns	
	$E_{test}$	$E_{test} - E_{train}$	$E_{test}$	$E_{test} - E_{train}$
Ferrites56	28,6 ± 3,6	6,8 ± 4,6	28,3 ± 3,5	6,0 ± 4,8
Ferrites32	32,6 ± 4,3	7,0 ± 5,3	32,1 ± 4,4	6,4 ± 4,4
Satimage72	20,6 ± 2,2	5,2 ± 3,0	20,4 ± 2,5	5,6 ± 2,8
Satimage36	24,5 ± 3,9	7,3 ± 5,2	24,1 ± 4,3	7,3 ± 5,0

## 6 Conclusion

It is no certain that error rate for real classification task is as optimistic as it is estimated during training phase. The value of the error rate determined in the proposed manner generates more confident estimation of the misclassification



probability. Generally, the method based on the analysis of the feature interdependences is recommended to be used, when the training set is small in relation to the number of features.

## References

1. Duda RO, Hart PE (2000) Pattern Classification (2nd ed.), Wiley
2. Chernoff H, Lehmann EL (1954) The use of maximum likelihood estimates in  $\chi^2$  tests for goodness-of-fit, *The Annals of Mathematical Statistics*
3. Rosner BA (2005) Fundamentals of Biostatistics, 6th ed., Duxbury Press
4. Tadeusiewicz R, Izvorski A, Majewski J (1993) *Biometria*, Wyd. AGH
5. Jozwik A, Chmielewski L, Sklodowski M, Cudny W (1998) A parallel net of (1-NN, k-NN) classifier for optical inspection of surface defects in ferrites, *Machine Graphics & Vision*, vol. 7, no. 1-2, pp. 99–112
6. Newman DJ, Hettich S, Blake CL, Merz CJ (1998) UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mlearn/MLRepository.html>], Irvine, CA: University of California, Department of Information and Computer Science

**Speech and Word Recognition**

---

# Normalisation of Confidence Voting Methods Applied to a Fast Handwritten OCR Classification

Juan Ramón Rico-Juan and José M. Iñesta

Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Alicante, E-03071 Alicante, Spain  
{juanra, inesta}@dlsi.ua.es

**Summary.** In this work, a normalisation of the weights utilized for combining classifiers decisions based on similarity Euclidean distance is presented. This normalisation is used by the confidence voting methods to decrease the final error rate in an OCR task. Different features from the characters are extracted. Each set of features is processed by a single classifier and then the decisions of the individual classifiers are combined using weighted votes, using different techniques. The error rates obtained are as good or slightly better than those obtained using a Freeman chain codes as contour representation and the string edit distance as similarity measure, but the complexity and classification time decrease dramatically.

## 1 Introduction

The combination of classifiers is a strategy widely applied recently in classification tasks. There are many ways to apply a combination of classifiers [1]. Methods based on decision confidences, like those reported in [2], allows to weight the individual classifier decisions in order to obtain a good classification performance.

There are some works which use confidence methods based in a posteriori probabilities (according to the Bayes theory) in classification, as in [3], where several formulas are proposed to estimate this probability, and in [4], where the authors propose a method based on the  $k$ -Nearest Neighbour rule.

In this work, we get into the details of the appropriate features, the individual classifiers, and their combination in order to classify isolated handwritten characters.

In the second section, we explain the different features that have been chosen for describing the character image. In the third section, the results obtained when applying different classifiers on a widely used database of isolated handwritten characters are shown. Finally, we present some conclusions and future lines of work.

## 2 Feature Extraction from a Binary Image

The goal of this feature extraction is to obtain different kinds of features to use with a specific classifier. Three different representations of the same image

(figure 1a) have been designed in order to obtain relevant information. The main idea is to summarise the image matrix information in proportional square regions that represent the character shape.

The first step is to locate the character in the image and extract the region of interest (ROI) as the character bounding box. This ROI is where the features described below are extracted from.

## 2.1 Foreground Features

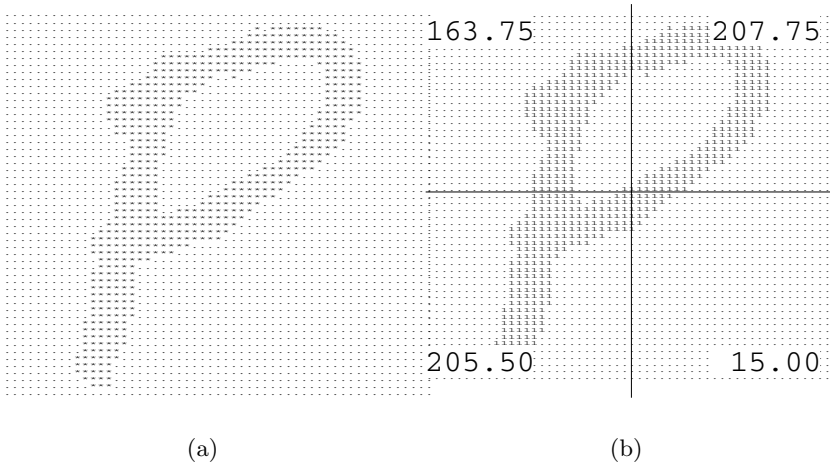
A classical representation of a bitmap image is a matrix with the black pixels as the smallest portions of the object ( $I[x, y] = 1$ ) and white pixels as the background ( $I[x, y] = 0$ ).

First, morphological filters [5] have been applied to correct gaps and spurious points. Thus, we define square subregions for representing the character ROI. Each subregion is represented by the number of foreground pixels. If a foreground pixel belongs to different regions, each region involved accumulates the proportional value of this pixel, as displayed in an example in figure 1b. The algorithm is detailed below, where  $I[I_{XM}, I_{YM}]$  is the bitmap image with  $I_{XM} \times I_{YM}$  dimensions and  $R[R_{XM}, R_{YM}]$  is the matrix with the proportional summatory of black pixels with  $R_{XM} \times R_{YM}$ .

```

1. function computeSummatoryOfRegions( $I[I_{XM}, I_{YM}]$ ,  $R[R_{XM}, R_{YM}]$ )
2.   for  $x := 1$  to  $R_{XM}$  do
3.     for  $y := 1$  to  $R_{YM}$  do
4.        $R[x, y] := 0$ 
5.       (startX, startY, endX, endY) := computeBoundingBoxObject( $I$ )
6.       for  $i :=$ startX to endX do
7.         for  $j :=$ startY to endY do
8.           if isBlackPixel( $I[i, j]$ ) then
9.             case  $[i, j]$  included in proportion  $[x, y]$  :
10.               $R[x, y] := R[x, y] + 1$ 
11.             case  $[i, j]$  included in proportion  $[x, y]$ ,  $[x, y + 1]$  :
12.               $R[x, y] := R[x, y] + \text{proportion}(x, y)$ 
13.               $R[x, y + 1] := R[x, y + 1] + \text{proportion}(x, y + 1)$ 
14.             case  $[i, j]$  included in proportion  $[x, y]$ ,  $[x + 1, y]$  :
15.               $R[x, y] := R[x, y] + \text{proportion}(x, y)$ 
16.               $R[x + 1, y] := R[x + 1, y] + \text{proportion}(x + 1, y)$ 
17.             case  $[i, j]$  included in proportion  $[x, y]$ ,  $[x, y + 1]$ ,
18.               $[x + 1, y]$ ,  $[x + 1, y + 1]$  :
19.               $R[x, y] := R[x, y] + \text{proportion}(x, y)$ 
20.               $R[x, y + 1] := R[x, y + 1] + \text{proportion}(x, y + 1)$ 
21.               $R[x + 1, y] := R[x + 1, y] + \text{proportion}(x + 1, y)$ 
22.               $R[x + 1, y + 1] := R[x + 1, y + 1] + \text{proportion}(x + 1, y + 1)$ 
22. end function

```



**Fig. 1.** a) Original image b) After morphological filters and a  $2 \times 2$  regions with the proportional summatory of the number of black pixels (object) for each subregion

## 2.2 Background Features

The algorithm used for extracting the background features is similar to that in [6]. This algorithm is based on four projections (up, down, left, and right) that are plotted for each pixel in the image. When any of these projections touch the foreground object, a counter associated to that pixel is increased in one unit. This way, we can distinguish four different categories of background pixels, according to the value of its counter. In addition, a fifth category is added in order to provide more information: there are two situations that are similar in geometry but totally different from a topological point of view. A background pixel can be surrounded by object pixels and then the projections will touch them and the counter will be 4, but this pixel could belong either to an isolated region or to an actually open region. So, our algorithm assigns a value of 5 to the counter if the pixel lies in an isolated background area.

Therefore, five matrices are extracted as features, one for each counter value. Each cell of these matrices represents the number of pixels with that counter value in the corresponding image subregion, as shown in an example in figure 2.

## 2.3 Contour Features

We extract the object contour encoding the links between each pair of neighbour pixels using 4-chain codes in the way proposed by [7], where only the orientations of the links are taken into account (see figure 3 for an example). We extract four matrices, one for each direction. In a similar way as we do in previous subsections, each cell of these matrices represent the summatory of each region for a direction as we show in an example in the figure 3a.

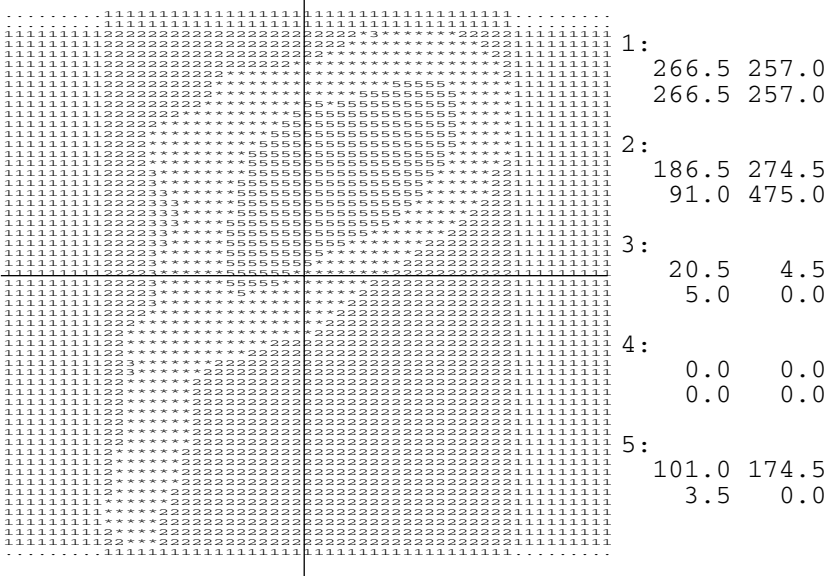


Fig. 2. Background features

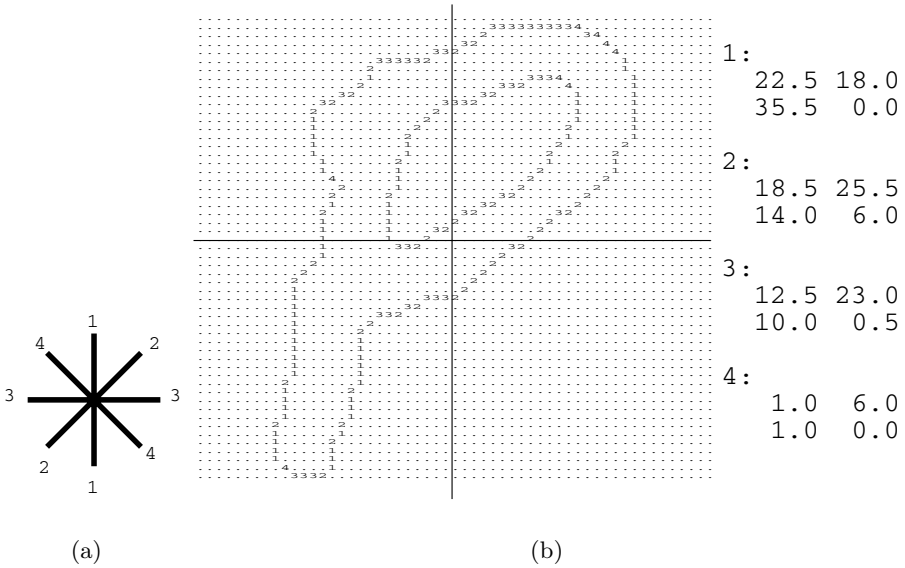
All these regions are normalised using  $p'_{ij} = p_{ij} / \sum p_{ij}$  to obtain better performance where  $p_{ij}$  is the cell  $i, j$  which represents the proportional summatory of the features in this region.

### 2.4 Classification Schemes

Three kinds of algorithms have been compared based on different features descriptions:

1. Individual classifiers based on the three sets of features described above. The features used are based in the image pixels (image), in the background information (background) and in the four directions of the countours image (directions).
2. Combination of classifiers based on confidence voting methods [2]:
  - Pandemonium[8]: Every classifier gives one vote with a confidence for every class. The class which receive the highest confidence is the winner.
  - Sum rule: Each class sums the confidence from each classifier. The class that accumulates the highest confidence is the winner.
  - Product rule: It is equivalent to *sum rule*, but the product is used for obtaining the final confidence for each class.

The confidences of the single classifiers are based on the Euclidean distance to the training set samples. We use a modification of the expression to compute *a posteriori* probability based on [3] and proposed in [4]. The estimation is



**Fig. 3.** a) Four 2D directions b) Contour encoded and a  $2 \times 2$  matrix with the summatory of number of pixels for each direction

$\hat{P}(\omega_i|x)$  where  $\omega_i$  is the  $i$ -th class and  $x$  is a new example to classify. The new estimation is computed as

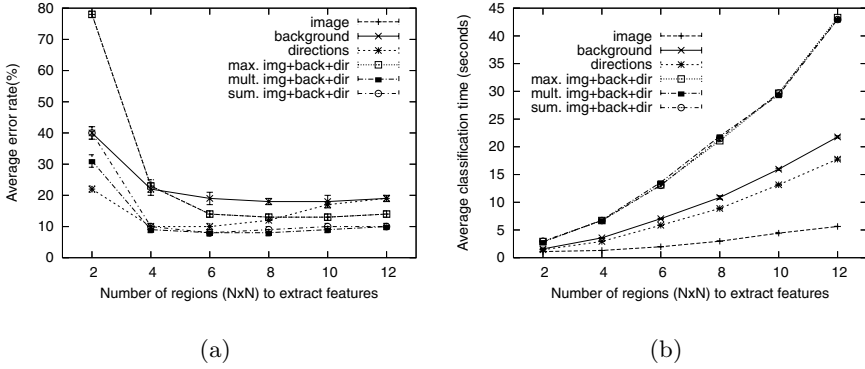
$$\hat{P}(\omega_i|x) = \frac{\frac{1}{\varepsilon + \min_{y_i \in \omega_i} \{d(x, y_i)\}}}{\sum_i \frac{1}{\varepsilon + \min_{y_i \in \omega_i} \{d(x, y_i)\}}}$$

This estimation is based on the nearest neighbour sample. A  $\varepsilon$  value is introduced that is a positive small value close to 0. It allows to apply this formula without overflow errors.

3. Contour of image as string using 8-Freeman codes as described in [9] (contour-string).

### 3 Results

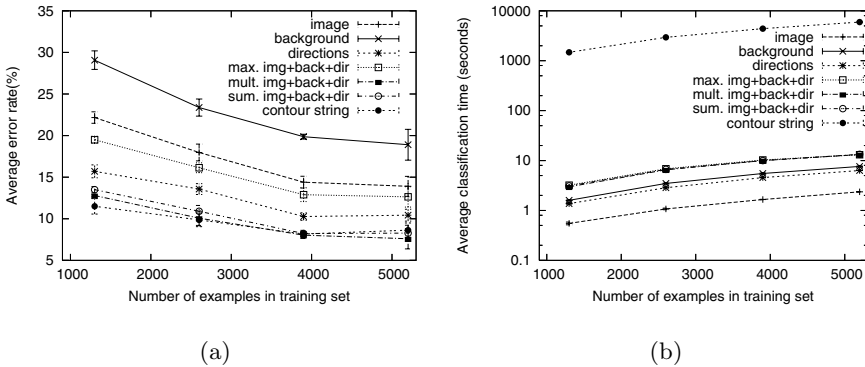
A classification task using the NIST SPECIAL DATABASE 3 of the National Institute of Standards and Technology was performed using the different contour descriptions described above to represent the characters. Only the 26 uppercase handwritten characters were used. The increasing-size training samples for the experiments were built by taking 500 writers and selecting the samples randomly. The size of the test is 1300 example in all cases, while the training set has different sizes as 1300, 2600, 3900 and 5200, corresponding to 50, 100, 150 and 200 examples per class.



**Fig. 4.** Preliminary trial tests with a 1300 test examples and 5200 model examples. a) Average classification error rate b) Average classification time.

As shown figure 4, in the preliminary trials tested, different numbers of regions were tested. A division of the ROI into  $6 \times 6$  subregions yielded a lower error rate than the others and the best computation time. Thus, the  $R_{XM}$  and  $R_{YM}$  parameters for the *computeSummatoryOfRegions* function was fixed to  $6 \times 6$  to compute the rest of experiments.

The sensitivity to the classifier ensemble architecture has been tested removing one of the three individual classifiers from the final combination, and the results were always worse than the combination of the three classifiers.



**Fig. 5.** a) Average classification error rate b) Average classification time

As we can see in the figure 5a, the lowest error rates were obtained by the combination of classifiers using the product rule. Anyway, this result is comparable to that of the combination of classifiers based on the summatory rule and the string contour classifier. In the figure 5b the classification time as a function of the number of samples is shown. Like in the case of the classifier ensemble,



the time is much lower than using contour chain codes, because chain codes use a string edit distance with a quadratic complexity and the average length of the strings is of 350 codes per character, while in the proposed approach the classifiers use a, much faster, Euclidean distance.

## 4 Conclusions and Future Work

A number of ways for shape feature extraction from a binary image have been presented. Each set of features was used for handwritten character recognition by different classifiers based on Euclidean metrics. In order to achieve a better performance, a weighted voting-based ensemble of those classifiers has been built. A posteriori probability estimation is proposed in order to normalize the confidences provided for each classifier in the voting phase for reducing the final classification error rate.

The classification time using these combination of classifiers was about 500 times faster than string edit distance-based classifiers operating with contour chain codes, obtaining even slightly better error rates.

This combination will be applied to other databases in the future to explore the robustness of the approach and how it performs in terms of both classification time and error rates with other data. In addition, we will explore its capabilities when dealing with other data in the biometric recognition field.

**Acknowledgement.** Work supported by the Spanish CICYT under project TIN2006-14932-C02, partially supported by EU ERDF and Generalitat Valenciana I+D+i under project GV06/166.

## References

1. L. I. Kuncheva. *Combining pattern classifiers : methods and algorithms*. John Wiley & Sons, 2004.
2. Merijn van Erp, Louis Vuurpijl, and Lambert Schomaker. An overview and comparison of voting methods for pattern recognition. In *IWFHR '02: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, page 195, Washington, DC, USA, 2002. IEEE Computer Society.
3. M. van Breukelen, D.M.J. Tax R.P.W. Duin, and J.E. den Hartog. Handwritten digit recognition by combined classifiers. *Computacional Linguistics*, 34(4):381–386, 1998.
4. J. Arlandis, J.C. Perez-Cortes, and J. Cano. Rejection strategies and confidence measures for a k-nn classifier in an ocr task. In *16th. International Conference on Pattern Recognition ICPR-2002*, volume 1, pages 576–579, Québec, (Canada), 2002. IEEE Computer Society.
5. J. Serra. *Image Analysis and mathematical morphology*. Academic Press, 1982.
6. E. Vellasques, L.S. Oliveira, A.S. Britto Jr., A.L. Koerich, and R. Sabourin. Modeling segmentation cuts using support vector machines. In *Tenth International Workshop on Frontiers in Handwriting Recognition*, volume 1, pages 41–46, 2006.

7. Hideto Oda, Bilan Zhu, Junko Tokuno, Motoki Onuma, Akihito Kitadai, and Masaki Nakagawa. A compact on-line and off-line combined recognizer. In *Tenth International Workshop on Frontiers in Handwriting Recognition*, volume 1, pages 133–138, 2006.
8. O. G. Selfridge. Pandemonium: a paradigm for learning in mechanisation of thought processes. In *Proceedings of a Symposium Held at the National Physical Laboratory*, pages 513–526, London, November, 1958. MIT Press.
9. J. R. Rico-Juan and L. Micó. Finding significant points for a handwritten classification task. In A. Campilho and M. Kamel, editors, *International Conference on Image Analysis and Recognition*, number 3211 in Lecture Notes in Computer Science, pages 440–446, Porto, Portugal, jun 2004. Springer.

---

# Extraction of the Visual Features from the Audio-Visual Speech Signal and the Utilization of These Features for the Speaker Identification

Josef Chaloupka

SpeechLab, Institute of Information Technology and Electronics, Technical University of Liberec, Czech Republic  
josef.chaloupka@tul.cz

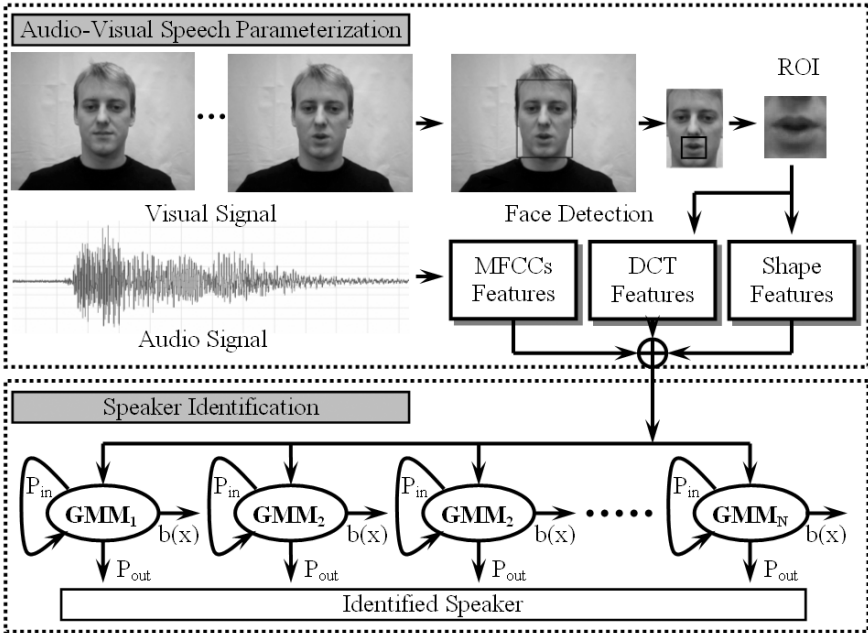
**Summary.** This paper is about the visual features extraction. The shape visual features and appearance based visual features were used in this work for the speaker identification. A Gaussian Mixture Models (GMMs) were trained from these visual features and the speakers were recognized with the help of these GMMs. A difference between the speaker identification with GMM trained from audio or visual features was investigated too.

## 1 Introduction

The human lips, teeth and mimic muscles have influence on the creation of the speech. The visual information of the speech can help hearing handicapped people to understand and the visual information is necessary for all people for understanding of the pronounced speech information in the noisy conditions. Therefore it will be good to use the visual information for the automatic speech recognition namely in the noisy conditions. The task of the audio-visual or visual only speech recognition consists of two steps. The audio-visual or visual signal (video recording of human speaker) is pre-processed and it is parameterized in the first step. The audio-visual or visual features are used in the automatic speech recognizer in the second step.

The speech recognizers are based on the Hidden Markov Models (HMMs) or on Artificial Neural Networks (ANNs) most often at present. The parameterization of the acoustic signal is well-solved - the mostly used audio features are the features based on the cepstrum. The Mel Frequency Cepstral Coefficients (MFCCs), which are extracted from the audio signal, are used for speech recognition, and the same features (without dynamic features) are used for speaker identification. We don't need to know a content of the speech utterance but it is necessary to find out the characteristic of speech for single speaker in the task of speaker identification.

The idea of this work was to use the visual features for the speaker identification and find out if it is possible to use the same visual features, which were used in our previous work for the audio-visual speech recognition [1]. The extraction



**Fig. 1.** The principle of speaker identification from the visual or audio speech signal by the help of GMMs technique

of the MFCCs is well-known and it is described in a large number of papers therefore the possibilities of the visual speech parameterization are described in this paper.

## 2 The Visual Features Extraction

The bottom part of the human face from the video image is used for the extraction of the visual features most frequently. We have two groups of the visual features at present. The "shape" visual features are extracted from the segmented object of the lips in the first group and the "appearance based" visual features are in the second group. The appearance based visual features are created with the help of the 2D transforms (DCT - Discrete Cosine Transform, DWT - Discrete Wavelet Transform, PCA - Principal Component Analysis, ...) of the region of interest (ROI). The lips and the lips environment are in the ROI. The lips are necessary to be found for the extraction of the shape or appearance based visual features. Direct lips finding in the real video image is very difficult task because the size of the lips is relatively small in comparison to the whole image and the real image is often very detailed. Therefore it is necessary to detect the human face(s) in the first step, and the lips are looked for in the bottom part of detected face. The video camera was located directly to human lips in

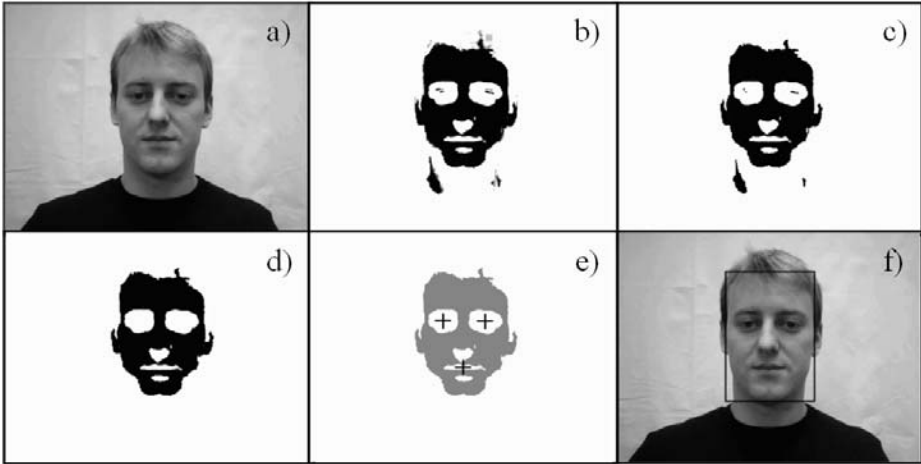
some older published works. The algorithms for the face detection and for the finding of the ROI didn't have to be used. This simplification is applicable only in the laboratory or simple conditions where the human speaker is not moving and it is not useful in the real conditions.

## 2.1 Face Detection

The solution of the face detection task is relatively difficult. A several people can be in one video image. The people can show their faces to the video camera from different angles, and they can be in different distances from the camera, thereby the areas of the human faces have different sizes. The lighting may also change during making of the video recording. The front-end pictures of the human face are used most often because it is more complicated to detect the face from the profile from the video images and it is very difficult to extract good visual features from the face profile. A lot of methods and algorithms have been described and created for the face detecting. It is possible to divide these methods to three groups.

The feature-based methods are in the first group. The information about the shape, skin colour, and the texture of the face are used in these methods very often. The methods with template matching are in the second group, and appearance-based methods are in the third group. The appearance-based methods are based on the eigenfaces, distribution, neural networks, support vector machines, sparse network of winnows, or hidden Markov models [2]. The methods from the second and third group are mostly very time-consuming, and we need to detect the face(s) at least from 20 images that we get from one second of the visual signal. Therefore it is necessary to have some relatively fast and robust method for the face detection.

The algorithm for face detection based on the colour and shape segmentation was developed in our lab. The face(s) is(are) detected within 20ms (in 3GHz PC) in video image with sizes 640 x 480 pixels with the help of this algorithm. The principle of the algorithm is the following. The RGB colour video image is segmented with the help of 3D look-up table that converts the RGB value to value 0 or 1 where 1 is used if the colour is similar to the colour of the human face and 0 is used in the other cases. The look-up table was created from our extensive image database with the skin pixels, and the utilization of look-up table is a very fast operation because in one step we get the segmented picture. The result is the binary picture where the binary noise is very often, therefore the morphologic image operation of opening with a small structure element is used for the filtration of the binary noise. The second step is the image operation called region identification. The objects are selected in this operation, and some elongated objects are removed, because it is unlikely that these objects are the areas of the human faces. In the last step the objects of eyes and lips are looked for in the selected objects. The area is selected as a face area if two objects of eyes and one object of lips are found in the predefined positions. The bottom part of the face area is used as the ROI.



**Fig. 2.** The face detection algorithm: a) original picture, b) binary picture, c) filtered picture, d) after region identification, e) eyes and lips tracking, f) resulting detected face

## 2.2 Visual Front End Features Extraction from the ROI

A lot of methods exist for visual features extraction at present. The majority of methods are based on the visual features extraction where the human faces are camera scanned from the front but new methods are using the algorithms for the visual feature extraction from the approximated 3D space with speaker(s) [3]. The approximated 3D space is created from two video images from two cameras (stereovision), or the mirror system and one video camera is used. The problem is that it is very time consuming to extract visual features from 3D space on a "normal" PC. We have two main groups of the visual features today (in 2007) - the shape visual features and the appearance (video pixels) based visual features. The shape visual features are extracted from the found contours of the lips objects. The most often used shape visual features are: horizontal opening of the lips, vertical opening of the lips, area of the lips and lips rounding [4].

These visual shape features were used in this work and the features were obtained from the object of the lips, which was segmented from the ROI with the help of algorithm for colour and shape segmentation. This algorithm is very similar to our algorithm for the face detection (without the last step), but the threshold for the image segmentation is automatically selected from the algorithm based on the estimation of the image histogram from the ROI [5]. It is possible to get the other shape visual features from the analysed contour border of the lips [6] where orientation of the lips contour edge is monitored or the upper and bottom part of the lips is approximated with a suitable curve, for example

by parabola. The shape visual features have very good interpretation but it is very difficult to find these features in the real video images with image noise because it is necessary to reliably find the contours of the lips for the creation of the shape features.

The appearance-based features are based on using of the information obtained from the pixels from the ROI. It is possible to use the pixels directly but it isn't so practical because, for example, for the ROI with the sizes 128 x 128 pixels we get 16,384 features. A lot of useless information is in these features, and it will be very difficult to train some good models from these features, therefore the image of the ROI is transformed with the help of some image transformation and only some components are selected as the features from the transformed space. The most used 2D transforms are: the Discrete Cosine Transform - DCT [7] or the Principal Component Analysis - PCA [8]. Some other transforms are used sometimes, for example the Discrete Wavelet Transform - DWT. We have used the DCT for the extraction of the visual features in our work [1]. The coefficients  $F(u, v)$  were computed from the DCT of ROI with the lips.

$$F(u, v) = \frac{2c(u)c(v)}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m, n) \cos\left(\frac{2m+1}{2N}u\pi\right) \cos\left(\frac{2n+1}{2N}v\pi\right) \quad (1)$$

where  $f(m, n)$  are the pixel values from ROI image and  $c$  are the coefficients:

$$c(k) = 1/\sqrt{2} \text{ for } k = 0 \text{ or } c(k) = 1 \text{ for } k > 0 \quad (2)$$

These coefficients were squared (the energy were computed from the coefficients) and the  $N$  highest energy coefficients from the DCT were used as the visual features.

### 3 Speaker Identification

The Gaussian Mixture Models - GMMs are used for the speaker identification very often, and we have a good experience with this technique for speaker identification in our lab, therefore they were used in this research. The GMM is one-state HMM but usually a larger number of mixtures  $M$  is used in the GMM than in the HMM. The output function  $b$  of the GMM state is:

$$b_s(\mathbf{x}) = \sum_{m=1}^M c_{sm} \frac{1}{\sqrt{(2\pi)^P \det \Sigma_{sm}}} \exp\left[-0.5(\mathbf{x} - \mathbf{x}_{sm}^-)^T \Sigma_{sm}^{-1}(\mathbf{x} - \mathbf{x}_{sm}^-)\right] \quad (3)$$

where  $\mathbf{x}$  is the vector of features,  $P$  is the number of features,  $\mathbf{x}_{sm}^-$  is the vector of mean values,  $\Sigma$  is the covariance matrix and  $M$  is the number of mixtures.

One GMM is trained for each speaker. The unknown parameterized speech signal from the unknown speaker is compared with the GMMs and the speaker

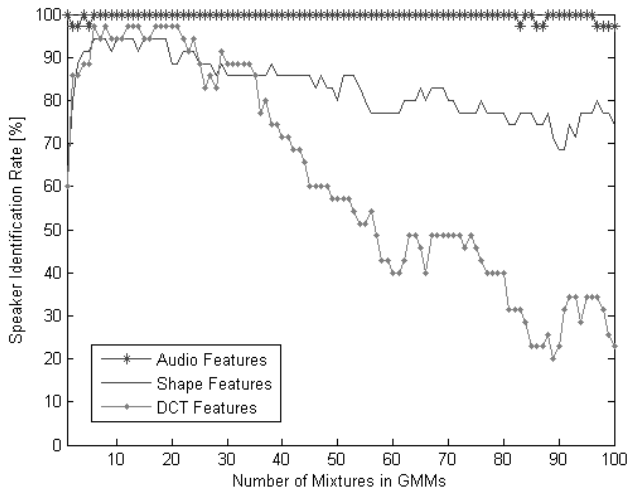
is identified as known where the resulting probability from the comparison is the highest. The unknown speaker should be in our database of GMMs, or it is necessary to decide if the speaker has the GMM in the database or not. The setting of the threshold for the comparison is the first way, but it isn't so good because it is very difficult to set the threshold, because if the threshold is set badly, it is possible that the speakers who have GMM in the database are selected as unknown. Using of the algorithms and methods for the speaker verification [9] is the second (and better) way, but only the algorithm for speaker identification from the visual or audio stream was used in this work.

## 4 Experiments

Our own Czech audio-visual speech database AVDBcz2 was used for the experiments of the speaker identification. This database was used in our previous works for the visual or audio-visual speech recognition therefore it was possible to use the same features that were already extracted from the audio-visual streams. 35 human speakers were camera-scanned in this database and each speaker narrated cca 60 seconds of utterances. The video frame rate is 30 images per second, the images are coloured with the sizes 640 x 480 pixels and the audio sample rate is 16 kHz in this database. The uncompressed database has cca 60 GB.

30 seconds of speaker's utterances were used for the training of the GMMs and the rest (cca 30 seconds) was used for the identification. The audio-visual signal was divided into frames - short parts of signal with the time length 33.3 ms and the frames were parameterized - this is the same strategy as the parameterization of audio signal for the speech recognition. 12 MFCCs were used for the parameterization of the audio frames and 4 shape visual features (horizontal opening of the lips, vertical opening of the lips, area of the lips, and lips rounding) or 5 highest energy DCT features were obtained from the visual frames. The GMMs were trained for each speaker from the audio or visual parameterized frames in the second step. The speakers were identified in the third step. The HTK Toolkit [10] was used for the training of the GMMs and for the speaker identification. 3 experiments for the speaker identification were created. The audio features were used in the first experiment, the shape visual features were used in the second experiment, and the energy DCT features were used in the third experiment. The number of mixtures  $M$  was changing in these experiments, see fig. 3. It wasn't possible to choose bigger (more than 100) number of mixtures for training of GMMs from visual features because this number depends on the number of data - the parameterized frames and therefore it is impossible to train the GMM's with bigger number of mixtures from our relatively small set of data. The resulting speaker identification rate was 100% (in big range of mixture values  $M$ ) for audio features, 97.1% ( $M = 6, 8, 12 \dots$ ) for DCT visual features and 94.3% ( $M = 6, 7, 8 \dots$ ) for shape visual features.





**Fig. 3.** The resulting speaker identification rate in dependence on number of mixtures in GMMs for the audio features, for the visual DCT features and for the shape visual features

## 5 Conclusion

The system for the speaker identification based on GMMs, which were trained from the visual speech features, was developed in this work. This work shows that it is possible to use the same visual speech features for speaker identification as for the speech recognition with a good speaker identification score. We have the same experience with the acoustic MFCCs features where the same static audio speech features are used for the speech recognition or for the speaker identification. The result, where the visual energy DCT features were used, is better than for the using of the shape visual features. It was the same for the visual speech recognition where the utilization of the energy visual DCT features gave better recognition rates. The main area for the using of the speaker identification from the visual stream is in the strong acoustic noise conditions where the speaker identification from the acoustic signal is either impossible or its result is very bad. We would like to extend our audio-visual database by new speakers in the near future, and we plan to create more extensive experiments to test how good visual features are for the speaker identification using a bigger group of speakers.

## Acknowledgements

The research reported in this paper was partly supported by the Czech Science Foundation (GACR) through the project No. 102/05/0278.

## References

1. Chaloupka J (2006) Creation and Selection of the Visual Front End Features and the Audio-Visual Feature Fusion for Audio-Visual Speech Recognition. In: Speech and Computer International Conference - Specom 2006, St. Petersburg, Russia, pp. 499-502, ISBN 5 7452 0074 x
2. Yang M H, Ahuja N (2001) Face Detection and Gesture Recognition for Human-Computer Interaction. In Kluwer Academic Publishers, USA, ISBN 0-7923-7409-6
3. Goecke R, Millar J B, Zelinsky A, Robert-Ribes JS (2001) Stereo Vision Lip-Tracking for Audio-Video Speech Processing. In Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP 2001, Salt Lake City, USA
4. Gao W, Ma J, Wang R, Yao H (2000) Towards Robust Lipreading, In International Conference on Spoken Language Processing. pp. 15-19, Beijing, China, ISBN 7 80150 114 4
5. Chaloupka J (2004) Automatic Lips Reading for Audio-Visual Speech Processing and Recognition. In: Proc. of ICSLP 2004, Jeju Island, Korea, pp. 2505-2508, ISSN 1225-441x
6. Kroschel K, Heckmann M (2002) Lip Parameter Extraction for Speechreading. In Conf. El. Sprachsignalverarbeitung, Dresden, pp.58-65
7. Duchnowski P, Meier U, Waibel A (1994) See me, hear me: Integration automatic speech recognition and lip-reading. In Proc. of ICSLP 94, Yokohoma, Japan, pp. 547-550
8. Bregler C, Konig Y (1994) Eigenlips for robust speech recognition. In Proc. of International Conference Acoustic, Speech and Signal Processing, Adelaide, Australia, pp.669-672
9. Silovsky, J, Nouza J (2006) Speech, Speaker and Speaker's Gender Identification in Automatically Processed Broadcast Stream. In: Radioengineering, Proceedings of Czech and Slovak Technical Universities and URSI Committees, Volume 15, Number 3, pp. 42-48, ISSN 1210-2512
10. Steve Y, Odel J, Ollason D, Valtchev V, Woodland P (1997) The HTK Book, version 2.1. In Cambridge University, United Kingdom

---

# Different Approaches to Class-Based Language Models Using Word Segments

Raquel Justo and M. Inés Torres

Dept. of Electricity and Electronics, University of the Basque Country, Spain  
webjublr@lg.ehu.es, manes@we.lc.ehu.es

**Summary.** In this paper we propose different approaches to the LM integrated in a Continuous Speech Recognition system. All of them are based on classes that are made up of phrases or segments of words. The proposed models were evaluated in terms of *Word Error Rate* over a spontaneous dialogue corpus in Spanish. The experiments carried out show that better performance of the CSR system can be achieved introducing segments of words into a class-based LM.

## 1 Introduction

Natural language modeling is a classical Pattern Recognition (PR) problem which is nowadays addressed within statistical PR framework. Word  $n$ -gram Language Models (LM) are the most commonly used approach, however, large amounts of training data are required to get a robust estimation of the parameters defining such models. When dealing with Continuous Speech Recognition (CSR) applications for which the amount of training material available is limited, alternative approaches as class  $n$ -gram LMs [1], could be used.

A class  $n$ -gram LM is more compact and generalizes better on unseen events. Nevertheless, it only captures the relations between the classes of words, while it assumes that the inter-word transition probability depends only on the word classes. This fact degrades the performance of the CSR system. To avoid the loss of information associated with the use of a class-based LM, other authors have proposed different approaches, e.g. model interpolation, aiming to take advantage of both the accurate modeling of word-based models for frequent events, and the predictive power of class-based models for unseen or rare events [1, 2, 3].

On the other hand, LMs based on phrases, or word segments, have already successfully been used within the speech recognition [4] and machine translation [5] frameworks. In this work a LM based on classes made up of segments [6] is employed in order to combine the benefits of word-based and class-based models. Two different approaches to class-based models are proposed and formulated. In both we make use of segments of words within the classes, in order to capture the relations between words in each class. Finally, two interpolated models are

presented as well. Such models make use of the proposed class-based models where classes are made up of segments of words. All the presented models are fully explained in Section 2.

Let us note that in this work *k-Testable in the Strict Sense* (*k-TSS*) models are used instead of the classical *n*-gram models. *k-TSS* languages are a subset of regular languages, thus, they can be inferred from a set of positive samples by an inference algorithm [7]. Actually, *k-TSS* LMs are considered as the syntactic approach of *n*-gram models, but this syntactic approach allows to integrate *K* *k-TSS* models (where  $k = 1, \dots, K$ ) into a unique Stochastic Finite State Automaton (SFSA) under a back-off smoothing [8]. Moreover, this formalism assists the integration of different LMs in a CSR system, as the proposals presented in this work require.

The proposed LMs were then integrated into a CSR system in a dialogue system application and evaluated in terms of *Word Error Rate* (WER). Several series of experiments were carried out on a spontaneous dialogue corpus in Spanish. These experiments show that the integration of word segments into a class-based LM yields a better performance of the CSR system.

## 2 Word Segments in Class-Based Language Models

Two different approaches to class-based LM-s are formulated below. Both of them are generated introducing segments or sequences of words inside the classes of a class-based LM. However, in the first approach,  $M_{sw}$ , the words in a segment are separately studied and the transition probability among them is calculated. In the second approach instead,  $M_{sl}$ , the words in a segment are linked and the whole segment is treated as a unique new “word” or lexical unit. Finally, two different linear interpolated models are presented.

### 2.1 LM-s Based on Classes of Word Segments, $M_{sw}$

Given a sequence of  $N$  words  $\bar{w} = w_1, w_2, \dots, w_N$ , let us define a segmentation ( $s$ ) of the sequence into  $M$  segments, as a vector of  $M$  indexes,  $s = (a_1, \dots, a_M)$ , such that  $a_1 \leq \dots \leq a_M = N$ . The  $\bar{w}$  sequence of words can be represented in terms of such segmentation as follows:

$$\bar{w} = w_1, \dots, w_N = w_{a_0=1}^{a_1}, \dots, w_{a_{M-1}+1}^{a_M=N} \quad (1)$$

where  $w_{a_{i-1}+1}^{a_i} = w_{a_{i-1}+1}, \dots, w_{a_i}$ .

On the other hand, let  $C = \{c_i\}$  be a previously defined set of classes, selected using any classification criteria. Each class consists of a set of segments previously defined as well. Each segment within a given class is made up of a sequence of several words. If the words in the sequence  $\bar{w}$  are classified using the  $C$  set of classes, the corresponding sequence of classes is written as  $\bar{c} = c_1, c_2, \dots, c_T$  where  $T \leq N$ .

In this work, only segmentations compatible with the possible sequences of classes ( $\bar{c}$ ) associated to each sequence of words, are considered. This set of segmentations is denoted by  $\mathcal{S}_{\bar{c}}(\bar{w})$ . That is, only segmentations having the following form will be considered

$$\bar{w} = w_1, \dots, w_N = w_{a_0=1}^{a_1}, \dots, w_{a_{T-1}+1}^{a_T=N} \quad (2)$$

where  $w_{a_{i-1}+1}^{a_i}$  must be a segment belonging to the  $c_i$  class.

The segmentation of a sequence of words can be understood as a hidden variable. In this way, the probability of a sequence of words  $\bar{w}$ , according to a LM based on classes made up of segments ( $M_{sw}$ ), can be obtained as:

$$\begin{aligned} P_{M_{sw}}(\bar{w}) &= \sum_{\forall \bar{c} \in \mathcal{C}} \sum_{\forall s \in \mathcal{S}_{\bar{c}}(\bar{w})} P(\bar{w}, \bar{c}, s) = \sum_{\forall \bar{c} \in \mathcal{C}} \sum_{\forall s \in \mathcal{S}_{\bar{c}}(\bar{w})} P(\bar{w}, s | \bar{c}) P(\bar{c}) = \\ &= \sum_{\forall \bar{c} \in \mathcal{C}} \sum_{\forall s \in \mathcal{S}_{\bar{c}}(\bar{w})} P(\bar{w} | s, \bar{c}) P(s | \bar{c}) P(\bar{c}) \end{aligned} \quad (3)$$

being  $\mathcal{C}$  the set of all the possible class sequences, given a predetermined set of classes  $\mathcal{C}$ .

The probability of a given sequence of classes,  $p(\bar{c})$ , can be calculated as a product of conditional probabilities. The probability of a  $c_i$  class given its history of classes ( $c_1^{i-1}$ ) is estimated using a  $k$ -TSS model, where  $k_c$  stands for the maximum length of the class history considered, as Equation 4 shows.

$$P(\bar{c}) = \prod_{i=1}^T P(c_i | c_1^{i-1}) \simeq \prod_{i=1}^T P(c_i | c_{i-k_c+1}^{i-1}) \quad (4)$$

The term  $P(s | \bar{c})$ , on the other hand, could be estimated using different approaches: zero or higher-order models, assuming that all the segmentations have the same probability, etc. Let us assume, for a matter of simplicity, the segmentation probability to be constant  $P(s | \bar{c}) = \alpha$ .

Finally,  $P(\bar{w} | s, \bar{c})$  is estimated in accordance with zero-order models. Thus, given a sequence of classes  $\bar{c}$ , and a segmentation  $s$ , the probability of a segment given a class  $c_i$  only depends on this  $c_i$  class, but not on the previous ones, as Equation 5 shows. The term  $P(w_{a_{i-1}+1}^{a_i} | c_i)$  represents the probability of a sequence of words, which must be a segment, given the class of this segment and is estimated using a  $k$ -TSS model as shown below.

$$P(\bar{w} | s, \bar{c}) \simeq \prod_{i=1}^T P(w_{a_{i-1}+1}^{a_i} | c_i) \simeq \prod_{i=1}^T \prod_{j=a_{i-1}+1}^{a_i} P(w_j | w_{j-k_{cw}+1}^{j-1}, c_i) \quad (5)$$

where  $k_{cw}$  stands for the maximum length of the word history that is considered in each class  $c_i$ .

Under this approach, several SFSA-s need to be integrated into the CSR system: a SFSA representing the transition probabilities among classes as well as one additional SFSA for each class, representing the transition probabilities among the words contained in the segments of the class. Moreover, acoustic models should also be integrated in the search network. A static full integration of all these models is computationally prohibitive, thus, each SFSA is integrated “on the fly” [9] in the search network only when needed.

**2.2 LM Based on Classes of Linked Words:  $M_{sl}$**

In a second approach, we propose a LM based on classes consisting of linked sequences of words. In this approach each segment,  $w_{a_{i-1}+1}^{a_i}$ , will be considered as a new lexical unit that cannot be divided into different words. Let us denote each lexical unit by  $l_i$ , where  $l_i \in \{\Sigma\}$ , being  $\{\Sigma\}$  the previously defined set of all the possible segments that will be inside the classes. The same hypothetical sets of segments and classes of 2.1 are considered here but assuming now that the segments cannot be separated in different words. Thus, a sequence of lexical units  $\bar{l} = l_1, \dots, l_M$  corresponds to a specific segmentation ( $s$ ) of the sequence of words  $\bar{w}$ .

$$\bar{w} = \underbrace{w_{a_0=1}^{a_1}}_{l_1}, \dots, \underbrace{w_{a_{M-1}+1}^{a_M}}_{l_M} \tag{6}$$

Assuming again that only segmentations compatible with a given class sequence ( $\bar{c} = c_1, \dots, c_T$ ) are to be considered; the possible sequences of lexical units, for a given sequence of words, will have the following form  $\bar{l} = l_1, \dots, l_T$ , where  $l_i$  is a segment belonging to  $c_i$ .

A sequence of lexical units involves a specific segmentation itself, thus, in this case,  $\bar{l}$  is considered as a hidden variable and the probability of a sequence of words is given by Equation 7.

$$\begin{aligned} P_{M_{sl}}(\bar{w}) &= \sum_{\forall \bar{c} \in \mathcal{C}} \sum_{\forall \bar{l} \in \mathcal{L}_{\bar{c}}(\bar{w})} P(\bar{w}, \bar{c}, \bar{l}) = \sum_{\forall \bar{c} \in \mathcal{C}} \sum_{\forall \bar{l} \in \mathcal{L}_{\bar{c}}(\bar{w})} P(\bar{w}, \bar{l} | \bar{c}) P(\bar{c}) = \\ &= \sum_{\forall \bar{c} \in \mathcal{C}} \sum_{\forall \bar{l} \in \mathcal{L}_{\bar{c}}(\bar{w})} P(\bar{w} | \bar{l}, \bar{c}) P(\bar{l} | \bar{c}) P(\bar{c}) \end{aligned} \tag{7}$$

being  $\mathcal{C}$  the set of all the possible class sequences, given a predetermined set of classes  $C$ .  $\mathcal{L}_{\bar{c}}(\bar{w})$  is the set of all the possible sequences of lexical units compatible with the given sequence of words and the possible sequences of classes.

The third term in Equation 7,  $P(\bar{c})$ , is estimated as stated in Equations 4 (see previous Section).

The second term in Equation 7 is the probability of a sequence of lexical units given a sequence of classes. Assuming again zero-order models, this probability is calculated as:

$$p(\bar{l} | \bar{c}) = \prod_{i=1}^T P(l_i | c_i) \tag{8}$$

A  $k$ -TSS model, with  $k = 1$ , i.e. an unigram, has been used to estimate this kind of probability for each class.

Finally, the first term in Equation 7,  $P(\bar{w} | \bar{l}, \bar{c})$  is equal to 1 when the sequence of lexical units,  $\bar{l}$ , and the sequence of classes,  $\bar{c}$ , are compatible with the sequence of words,  $\bar{w}$ , and 0 otherwise.

**2.3 Interpolating a  $M_{sw}$  Model and a Word-Based LM,  $M_{hsw}$**

The interpolation of a class-based and a word-based LM has demonstrated to outperform both mentioned models. Therefore, a hybrid model ( $M_{hsw}$ ) is defined, here, as a linear combination of a word-based LM,  $M_w$ , and a LM based

on classes made up of word segments,  $M_{sw}$ . Using such a model the probability of a word sequence is given by Equation 9.

$$P_{M_{hsw}}(\bar{w}) = \lambda P_{M_w}(\bar{w}) + (\lambda - 1)P_{M_{sw}}(\bar{w}) \tag{9}$$

In the above equation, the term  $P_{M_w}(\bar{w})$  is the probability of a word sequence using a classical word-based language model, and in this work, a  $k$ -TSS models was used to estimate this probability, as Equation 10 shows.

$$P_{M_w}(\bar{w}) = \prod_{i=1}^N P(w_i|w_1^{i-1}) \simeq \prod_{i=1}^N P(w_i|w_{i-k_w+1}^{i-1}) \tag{10}$$

The term  $P_{M_{sw}}$  is calculated using the equations given in Section 2.1.

### 2.4 Interpolating a $M_{sl}$ Model and a Phrase-Based LM, $M_{hsl}$

On the other hand, another hybrid model  $M_{hsl}$  is defined as a linear combination of the class-based LM presented in the second approach,  $M_{sl}$ , and a LM based on the already defined lexical units ( $M_l$ ), i.e. a phrase-base LM [10].

Using such a model the probability of a word sequence is given by Equation 11.

$$P_{M_{hsl}}(\bar{w}) = \lambda P_{M_l}(\bar{w}) + (\lambda - 1)P_{M_{sl}}(\bar{w}) \tag{11}$$

The term  $P_{M_l}(\bar{w})$  is the probability of a word sequence using the phrase-based language model, computed as Equation 12 shows.

$$P_{M_p}(\bar{w}) = \prod_{i=1}^T P(l_i|l_1^{i-1}) \simeq \prod_{i=1}^T P(l_i|l_{i-k_l+1}^{i-1}) \tag{12}$$

$k_l$  stands for the maximum length of the phrase history, where phrases are the new lexical units defined in Section 2.2.

The term  $P_{M_{sl}}$  is calculated using the equations given in Section 2.2.

## 3 Classes and Word Segments

In order to deal with the proposals presented in the previous Section, a set of segments and a set of classes formed by those segments were obtained from the selected corpus. Two different types of criteria were used.

**Linguistic classes and segments:** In this case, the set of segments and the set of classes were simultaneously obtained under a linguistic criterion by applying a rule based method. These classes are independent of the task and consist of word segments having the same linguistic function. This set of classes, as well as the segments the classes are made up of, were provided by *ametzagaina*<sup>1</sup>.

---

<sup>1</sup> Ametzagaiña R&D group, member of the Basque Technologic Network, <http://www.ametza.com>

Furthermore, they provided us with the segmented and classified corpus. A total of 57 classes and 3,851 different segments were obtained over the corpus described in the next Section.

**Statistical classes and segments:** In this case, we first obtained a set of segments using a statistical criterion. The most frequent n-grams of the corpus were selected as segments. In this sense, and in order to avoid rare or unimportant n-grams, a minimum number of occurrences was required. Then, a segmented training corpus was generated with the set of segments. Finally, different sets of statistical classes constituted by the defined segments were obtained with the aid of *mkcls* [11]. A total of 1,289 different segments were obtained using this technique over the corpus described below.

## 4 Task and Corpus

The experiments were carried out over a task-oriented corpus that consists of human-machine dialogues in Spanish. In this corpus, 225 speakers ask by telephone for information about long-distance train timetables, fares, destinations and services. A total of 900 dialogues were acquired using the Wizard of Oz technique. A total number of 8,606 training sentences were employed to train the LMs, where 5,590 sentences were different from each other. The number of different words was 77,476 whereas the size of the vocabulary was 856. The test set was made up of 1,348 sentences with a vocabulary of 503 words and 72 Out of Vocabulary words. This task has intrinsically a high level of difficulty due to the spontaneity of the speech and the problematic derived from the acquisition of large amount of transcriptions, of human-machine dialogues, for training purpose. Therefore, it is well-suited to study the improvements associated to modifications in the LM.

## 5 Experiments and Results

The LMs proposed in this work were fed into a CSR system, which was subsequently evaluated in terms of WER.

Four series of experiments were carried out in order to evaluate the proposed models ( $M_{sw}$ ,  $M_{sl}$ ,  $M_{hsw}$ ,  $M_{hsl}$ ). Making use of these LMs, different experiments were performed. First of all, the set of linguistic classes was employed. Then, five experiments were carried out with each model using 50, 100, 200, 300 and 400 statistical classes respectively. The parameters selected for each model are enumerated below:

- $M_{sw}$ : A value of  $k_c = 3$  and  $k_{cw} = 2$  was established.
- $M_{sl}$ : A value of  $k_c = 3$  was selected.
- $M_{hsw}$ : For the  $M_{sw}$  model involved: A value of  $k_c = 3$  and  $k_{cw} = 2$  again and for the  $M_w$  model: a value of  $k_w = 3$ .
- $M_{hsl}$ : For the  $M_{sl}$  model:  $k_c = 3$  and for the  $M_l$  model:  $k_l = 3$ .



**Table 1.** WER results for the proposed LMs ( $M_{sw}$ ,  $M_{sl}$ ,  $M_{hsw}$  and  $M_{hsl}$ ) and different sets of classes: 57 linguistic classes and 50, 100, 200, 300 and 400 statistical classes. The WER results for the classical word-based model ( $M_w$ ) and the two described phrase-based models,  $M_{l(stat.)}$  and  $M_{l(ling.)}$ , are also shown.

		WER (%)						
no. cat.		$M_{sl}$	$M_{sw}$	$M_{hsl}$	$M_{hsw}$	$M_w$	$M_{l(stat.)}$	$M_{l(ling.)}$
<b>ling.</b>	<b>57</b>	22.78	25.97	21.53	20.04	19.84	20.04	23.53
<b>statis.</b>	<b>50</b>	20.96	22.52	18.46	19.23			
	<b>100</b>	19.83	21.21	18.35	18.84			
	<b>200</b>	19.42	20.79	18.30	18.14			
	<b>300</b>	19.27	20.66	18.53	18.22			
	<b>400</b>	19.63	21.38	19.06	18.52			

For the two hybrid models the  $\lambda$  parameter was selected to obtain the best WER result ( $\lambda = 0.1$ ) in both cases.

Table 1 illustrates WER results using the proposed LMs and, for comparison purposes, the results obtained with the classical  $M_w$  (the same model employed in the  $M_{hsw}$  model) with a value of  $k_w = 3$  and the  $M_l$  models (with  $k_l = 3$ ). Two different  $M_l$  models were generated, one of them, using the linguistic segments ( $M_{l(ling.)}$ ), to be interpolated with the  $M_{sl}$  model that is generated with the linguistic set of segments and classes. The other one uses the segments obtained under the statistical criterion ( $M_{l(stat.)}$ ) and is interpolated with the  $M_{sl}$  model that uses statistical segments.

Comparing the results of Table 1 with the ones obtained in another work [12], over the same task and using a classical class-based model with classes made up of isolated words, it can be concluded that better results are obtained when using word segment based classes (in both  $M_{sw}$  and  $M_{sl}$  models).

Regarding the experiments carried out with the  $M_{sl}$  model, a significant drop of the WER is observed compared to the previous model ( $M_{sw}$ ) for all of the selected sets of classes. The best result is obtained for 300 statistical classes, achieving an improvement of a 6.7% with respect to the value obtained in the same conditions for the  $M_{sw}$  model. Furthermore, this result even improves the WER value obtained with the word based LM ( $M_w$ ) by a 2.8%.

However, the use of the two hybrid models, outperforms the results obtained with all the previous proposals. Moreover, the best result is obtained with the  $M_{hsw}$  model, for 200 statistical classes where an improvement of a 8.56% is observed with respect to the word-based LM.

## 6 Concluding Remarks

The experiments, carried out in this work, show that using a LM based on classes consisting of segments of words instead of a classical class n-gram LM, a better performance of a CSR system can be achieved.

**Acknowledgement.** We would like to thank the *Ametzagaiña* group and, in particular, Josu Landa, for providing us with the linguistic categorization and segmentation of the corpus. On the other hand, this work has been partially supported by the University of the Basque Country under grant 9/UPV00224.310-15900/2004 and by CICYT under grant TIN2005-08660-C04-03.

## References

1. Brown, P.F., Pietra, V.J.D., Souza, P.V.d., Lai, J.C., Mercer, R.L.: Class-based n-gram Models of Natural Language. *Computational Linguistics* **18**(4) (1992) 467–480
2. Niesler, T., Whittaker, E., Woodland, P.: Comparison of part-of-speech and automatically derived category-based language models for speech recognition. In: ICASSP'98, Seattle. (1998) 177–180
3. Zitouni, I.: Backoff hierarchical class n-gram language models: effectiveness to model unseen events in speech recognition. *Computer Speech and Language* **21**(1) (2007) 99–104
4. Deligne, S., Bimbot, F.: Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams. In: Proc. ICASSP '95, Detroit, MI (1995) 169–172
5. Marcu, D., Wong, W.: A phrase-based, joint probability model for statistical machine translation. (EMNLP), Philadelphia, PA, July 6-7 (2002)
6. Ries, K., Buo, F.D., Waibel, A.: Class phrase models for language modelling. In: Proc. ICSLP '96. Volume 1., Philadelphia, PA (oct 1996) 398–401
7. García, P., Vidal, E.: Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(9) (1990) 920–925
8. Torres, I., Varona, A.: k-tss language models in speech recognition systems. *Computer Speech and Language* **15**(2) (2001) 127–149
9. Caseiro, D., Trancoso, I.: Transducer composition for on-the-fly lexicon and language model integration. In: Proceedings ASRU'2001, Madonna di Campiglio, Italy (December 2001)
10. Kuo, H.K.J., Reichl, W.: Phrase-based language models for speech recognition. In: Proceedings of EUROSPEECH 99. Volume 4. (September 1999) 1595–1598 Budapest, Hungary.
11. Och, F.J.: An efficient method for determining bilingual word classes. In: EACL99, Bergen (1999) 71–76
12. Justo, R., Torres, M.I., Benedí, J.M.: Category-based language model in a spanish spoken dialogue system. *Procesamiento del Lenguaje Natural* **37**(1) (2006) 19–24

---

# Recognition of Ultra Low Resolution Word Images Using HMMs

Farshideh Einsele, Rolf Ingold, and Jean Hennebert

Université de Fribourg, Boulevard de Pérolles 90, 1700 Fribourg, Switzerland  
{farshideh.einsele,rolf.ingold,jean.hennebert}@unifr.ch

**Summary.** We report in this paper on significant improvements that we have been including in our HMM-based system for recognition of *ultra low resolution, anti-aliased text with small point sizes* such as those frequently found in web images. First we are proposing a fully automatic training procedure where no a priori knowledge of font metrics is needed. This means that our system can be potentially built to recognize any font. Second, the system's performance can be boosted by using mixtures of Gaussians to model the probability density functions of the HMMs. Third, we show that these improvements allow the system to handle large vocabulary size up to 60'000 words with few degradation of the accuracy. We report on these results for 2 different font families, namely a serif and a sans serif font. We also report on different HMMs topologies and conclude on the benefits of using minimum duration to model the characters composing the words.

## 1 Introduction

The tremendous growth of the World Wide Web has proven to be a big challenge in the field of information retrieval and indexing. According to a study done in 2001 [1], of the total number of words visible on a WWW page, 17% are in image form and 76% of these words do not appear elsewhere in the encoded text. Such embedded text contains often important information with a high semantical value that could be beneficial for indexing technologies.

Text embedded in web images is often anti-aliased with small font sizes (< 12 pts) and has *ultra* low resolution (between 72 and 90 dpi). Classical OCRs are built to treat high-resolution (>150 dpi) bi-level images acquired from scanned documents and are therefore not designed to recognize such text. For this reason, several works [2] [3] [4] have been proposing pre-processing methods to reduce the mismatch between text embedded in web images and the quality expected by classical OCR systems. These works have been reporting noticeable successes.

Instead of trying to pre-process the text image to feed it into a classical OCR, we are proposing in this paper to use a recognizer specifically trained to recognize such inputs. To achieve this, we propose to use Hidden Markov Models (HMMs) that offer robust training methodologies and the ability to

incorporate lexical or even grammatical constraints. Finally, HMMs are versatile tools, applied in many different fields for quite some time such as in the field of cursive handwriting [5] or automatic speech recognition [6]. They have also been used to improve the robustness of classical OCR systems to make them language or script independent [7] [8] or even to boost the accuracy [9]. However, to the best of our knowledge, using a HMM based recognizer specifically trained to recognize *ultra low resolution, anti-aliased words with small point sizes* is a novel approach.

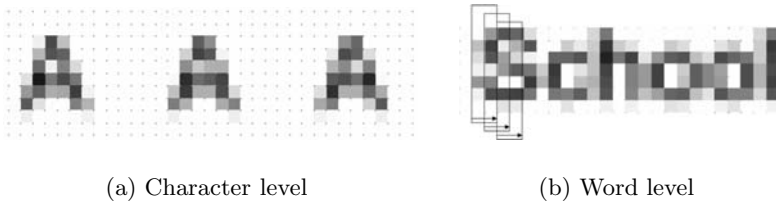
In our previous work, we have been showing the proof of concept of the approach by conducting a preliminary study on isolated characters [10]. The focus was on the understanding of the variabilities of text in web images and on identifying a reliable feature extraction and local character scoring. We have shown that a feature extraction based on moments computation and on multivariate Gaussian density functions leads to robust results. In [11], we moved to use HMMs to recognize words. In this work, the training of the HMMs parameters was performed on low resolution word images where the segmentation of characters was obtained in a knowledge-based manner, inferring it from some font metric information.

We report in the present paper on three major improvements that we have been including in our system. (1) We do not rely anymore on any a priori knowledge of font metrics to train the HMMs parameters, i.e. we let the HMMs discover the segmentation during the training procedure, making the system able to train on unseen fonts. (2) We relax the assumption of mono-Gaussian distribution of the extracted features to move to multiple Gaussian mixture distributions. (3) Thanks to the previous improvements, we were able to increase significantly the vocabulary size up to 60000 words. The organization of this paper is as follows. In section 2 we discuss the specificities of *ultra low resolution text* images. In section 3 we present our HMM-word recognizer system and the related improvements that are introduced. In section 4 we report on the performance of the system with increased vocabulary size and for two different font families. The paper ends with conclusions and future works.

## 2 Specificities of Text in Web Images

**Character level.** As shown on Fig. 1(a), a character embedded in a web image presents specific characteristics: (1) the character has an *ultra* low resolution, usually smaller than 100 dpi with small point sizes frequently between 6 and 12 points, (2) the character has artefacts due to anti-aliasing filters (3) the same characters can have multiple representations due to the position of the down sampling grid.

**Word level.** Fig. 1(b) illustrates a realistic example of characters composing the word 'School' in a low resolution image. As can be observed, there are no spacings available to segment characters within the word. Therefore the well-known pre-segmentation methods [12] used in classical OCR systems can not be applied anymore in our case. Furthermore, the anti-aliasing noise on both

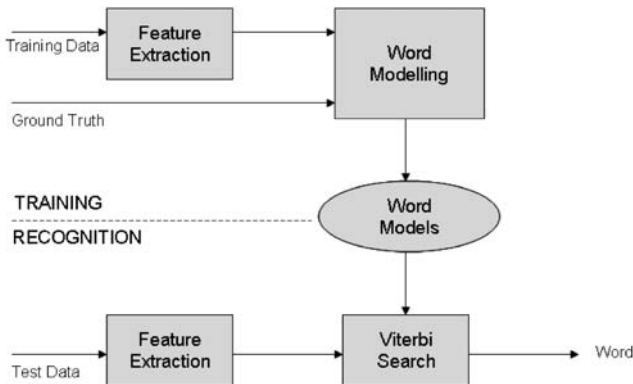


**Fig. 1.** (a) Example of anti-aliased, down sampled character 'A' with different grid alignments. (b) Low resolution version of word 'School' and illustration of the sliding window used for the feature extraction.

borders of adjacent characters is also source of variability. Acknowledging this, we opted to use a HMM based recognizer that is able to solve simultaneously the segmentation and the recognition problem, being able to cope with the noise between the adjacent characters.

### 3 System Description

Our web-image-OCR system is currently a single word recognizer and not a connected word recognizer. We can indeed reasonably assume that words can be isolated with classical image processing techniques, even though they are low resolution and have small font sizes. However, we do not discard at this stage the use of connected word recognizers that could benefit from the inclusion of grammatical constraints on sequence of words. As illustrated in the block diagram of Fig. 2, our system consists of two processes, namely the training and the recognition system. Both systems are based on HMMs where states are associated to characters. In this manner, any word can be modelled by a HMM where the corresponding states are simply connected together.



**Fig. 2.** Block diagram of the Web-Image-OCR

### 3.1 Feature Extraction

HMMs model ordered sequences of features that are function of a single independent variable. In our presented work, we decided to compute a left-right ordered sequence of features by sliding an analysis window on top of the word. Therefore the independent variable is in our case, the x-axis. Since *few pixels* are available for each character, we decided to use the *first and second order central moments*, since they have proven to be fairly discriminative in our previous studies [10] [13]. As illustrated on Fig. 1(b), we used a 2 pixels length window shifted 1 pixel right. In each analysis window, we compute a feature vector of 8 components including the 6 first and second order central moments, the weight (sum of gray pixel values) and the baseline of each character. The word image is then represented by a sequence of feature vector  $X = \{x_1, \dots, x_N\}$  where  $x_n$  is a 8 component vector.

### 3.2 Training Method

The training method is performed directly on words using an expectation maximization (EM) iterative process (see for example [6]). The likelihood-function we selected is based on continuous Gaussian mixtures. Additionally, we have made the assumption that the components of the feature vector are uncorrelated. This presents the advantage to let the covariance matrix be diagonal and to be more computationally efficient. We have measured that this assumption is actually not critical in terms of accuracy [11]. The EM algorithm iteratively refines the HMM parameters to increase monotonically the likelihood of the observed feature vectors given the model.

### 3.3 Word Models

We have decided to use the HMMs to model the words. The HMMs model the likelihood of the observation sequence  $X$  given the model parameters associated to a word  $i$ . By applying the usual simplifying assumption of HMMs (see for example [6]), the likelihood  $P(X|M_i)$  of  $X$  given the model  $M_i$  can be expressed as the sum, over all possible paths of length  $N$ , of the product of emission probabilities and transition probabilities measured along the paths. Alternatively, the Viterbi criterion can also be used, stating that instead of considering all potential paths through the HMM, only the best path is taken into account, i.e. the path that maximizes the product of emission and transition probabilities. In this work, we have based our training and testing approaches using the Viterbi criterion. As stated before, the characters are associated to states in our word models. The number of states is defined by the number of characters composing the word and the choice of the HMM topology to model each character. We have chosen here to model the characters using two different HMM topologies that are described hereafter. We also introduce an additional pseudo-character state that models the noise between the adjacent characters in the words.

**Inter-character model.** According to our previous study of characters in context (see [13]), the left and right borders of characters are influenced by their adjacent characters. Our experiments have shown that this *noisy* zone spans from one to three pixels depending on the adjacent character shapes, sizes and font families. We have chosen to train a specific model using these inter-character zones, treating it as an extra character in our framework. This new pseudo-character is called the *inter-character* ( '#' in our figures). Fig. 3 shows an example of the resulting HMM topology for the word 'zero'. As illustrated, the inter-character models have the possibility to include one or two associated feature vectors which correspond to a maximum span of three pixels.

**Simple left-right topology.** As illustrated on Fig. 3, the words are modelled by connecting every composing character models. Characters are associated to one single state with the possibility of an unlimited number of self transitions. The inter-character model is inserted between each adjacent character.

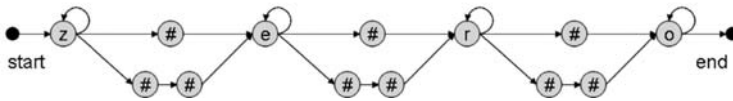


Fig. 3. Simple left-right topology for word 'zero'

**Minimum duration topology.** This topology is simply obtained by repeating a state a given number of time, proportionally to the character length. Doing this, the decoding Viterbi algorithm is forced to spend a minimum number of feature vectors in a given character. Confusions between characters including similar repeated patterns will then be better avoided, such as for example between letters 'n' and 'm'. In our configuration, the minimum duration values are obtained from the a priori known font metric information.

### 3.4 Testing Framework

At recognition time a large HMM is built by taking into account the vocabulary defined in a dictionary and according to a tree-like structure to optimize the memory. In order to reduce computation time, a classical pruning procedure is applied to discard the less probable paths along the dynamic programming computation. As we are testing our system on unseen vocabulary, the HMMs for each word are composed using the states trained on each character. In our approach, we do not use the transition probabilities obtained at training time as their influence have been measured to be negligible.

## 4 Experimental Results

We train our system using 3000 synthetically generated word images. First the images are produced in higher resolutions using the `java.awt.Font` class and are

then down sampled and anti-aliased using Photoshop to our target resolution. We also focus on single font recognition in this work, which means that each font is modelled and tested independently.

**Table 1.** Influence of the number of Gaussians on the recognition rate of a 3'000 word system for two font families, Plain, 9 pts

# Gaussians	Sans Serif	Serif
1	52.6%	16.6%
2	99.5%	97.5%
4	99.0%	98.9%

due to the fact that serif fonts have more complicated character shapes including small flourishes. We can also observe that the serif font requires more complex models than for the sans serif font. Using 4 Gaussian mixtures seems to be more accurate for serif while 2 seems to be optimal for the sans serif. These results indicate that there could be an optimization of the number of mixtures for each font and probably also for each character but we did not investigate this in this work.

Our second set of experiments is summarized in Table 2. We analyze here the influence of the vocabulary size and of the different HMM-topologies described above. Results were obtained using the 2 Gaussian mixtures system. We see that the accuracy is slightly decreasing when increasing the vocabulary size. However, the system is still leading to good accuracy even for large vocabulary up to 60'000 words. As it was already observed in our previous work, results also show that the minimum duration topology delivers in general better results. As stated in the first set of experiments, we also see that the serif font is more difficult to recognize than the sans serif font.

**Table 2.** Recognition rates for different vocabulary size and HMM topologies and for two font families, **sans serif** (serif), Plain, 9 pts, 2 Gaussian mixtures

	3'000 words	12'000 words	60'000 words
Simple left-right	<b>99.5%</b> (97.5%)	<b>98.0%</b> (92.4%)	<b>96.0%</b> (90.3%)
Minimum duration	<b>99.8%</b> (98.6%)	<b>99.4%</b> (96.5%)	<b>97.7%</b> (93.4%)

We also performed an error analysis that is summarized in Table 3 for the serif font. Many errors are related to confusions between characters that are globally similar such as for example 't' and 'i' or 'o' and 'e'. Increasing the number of Gaussian mixtures allows to recover some of these mistakes by allowing more precise modelling. This is clearly illustrated for the word 'aloi' and 'wenda'. Besides, a large number of errors are related to characters that share local similarities such as, for example, the vertical strokes of letters 'm' and 'n'. This also

We first investigate on the effect of the number of Gaussian mixtures to model the emission probability function in each state. We have performed different experiments using 1, 2 and 4 mixtures. Results are summarized in Table 1 for a sans serif and a serif font. We see clearly that increasing the number of Gaussian mixtures from 1 to 2 drastically increases the recognition rates. Going from 2 to 4 mixtures does not lead to much bigger improvements. Compared to the sans serif font, the serif font system reaches less accuracy. This is probably



lead to insertion errors as in the case of 'd' recognized as 'cl'. The introduction of minimum duration topology allows to recover some of these mistakes. In some examples, this is the combination of minimum duration and model complexity that allows to remove the mis-recognition, like for the word 'shah' in Table 3.

**Table 3.** Typical examples of mis-recognition for sans serif font, 12'000 test words

Genuine word	Recognized word			
	simple, 2-Gauss	simple, 4-Gauss	min dur, 2-Gauss	min dur, 4-Gauss
aloi	alot	aloi	alot	aloi
wenda	wonda	wenda	wonda	wenda
moller	noller	noller	moller	moller
dumm	clum	clum	clum	dumm
shah	shale	shale	shale	shah

## 5 Conclusion and Future Work

We have reported on improvements and performance evaluation of our *web-image-OCR* that is specifically trained to model and recognize *anti-aliased, ultra low resolution text with font sizes up to 12 points*. The novelty of our approach is to use a recognizer specifically trained to recognize such inputs instead of using a classical OCR with some preprocessing of the text image. The system is based on an isolated HMM-word recognizer. We have shown in this paper that the system is able to automatically train on non-segmented data and to recognize up to 60'000 words with fairly good accuracy (97.7%). Key elements of the HMM system that have been leading to improvements of the accuracy include the use of inter-character models, the use of more complex probability density functions based on Gaussian mixtures and the use of minimum duration topologies. We could also show that our system can be automatically trained on other font families, still leading to similar performances. We noticed that serif fonts are leading to a slight degradation of the accuracy probably due to their additional decorative marks. In our future work we are planning to further enhance the system at different levels such as, for example, improving the feature extraction algorithm to include more contextual information in the analysis window or making the modelling more precise by introducing a number of Gaussian mixtures dependent to the character complexity. In addition to this, we intend to move towards multi-font systems and to open dictionary using ergodic HMMs.

## References

1. Antonacopoulos, A., Karatzas, D., Lopetz, J.: Accessing textual information embedded in internet images. In: Proceedings of Electronic Imaging, Internet Imaging II, San Jose, California, USA (Jan. 2001)
2. Lopresti, D., Zhou, J.: Locating and recognizing text in www images. In: Information Retrieval, 2 (2/3). (May 2000) 177–206

3. Antonacopoulos, A., Karatzas, D.: Text extraction from web images based on a split-and-merge segmentation method using color perception. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR2004), Cambridge, UK (August 23-26, 2004)
4. Perantonis, S., Gatos, B., Maragos, V.: A novel web image processing algorithm for text area identification that helps commercial ocr engines to improve their web recognition accuracy. In: Proceedings of the second International Workshop on Web Document Analysis, Edinburgh, United Kingdom (August 3, 2003)
5. Marti, U.V., Bunke, H.: Using a statistical language model to improve the performance of a hmm-based cursive handwriting recognition system. In: *Int. Journal of Pattern Recognition and Art. intelligence*, 15. (2001) 65–90
6. Rabiner, L., Juang, B.H.: *Fundamentals Of Speech Recognition*. Prentice Hall (1993)
7. Lu, Z., Bazzi, I., Kornai, A., Makhoul, J., Natarajan, P., Schwartz, R.: A robust, language-independent ocr system. In: Proceedings of SPIE, 27th AIPR Workshop: Advances in Computer-Assisted Recognition. Volume 3584. (January 1999) pp. 96–104
8. Lu, Z., Schwartz, R., Raphael, C.: Script independent, hmm-based text line finding for ocr. In: Proceedings of SPIE, 27th AIPR Workshop: Advances in Computer-Assisted Recognition. Volume 3584. (January 2000) pp. 96–104
9. Velagapudi, P.: Using hmms to boost accuracy in optical character recognition. In: Proceedings of SPIE, 27th AIPR Workshop: Advances in Computer-Assisted Recognition. Volume 3584. (January 1999) pp. 96–104
10. Einsele, F., Ingold, R.: A study of the variability of very low resolution characters and the feasibility of their discrimination using geometrical features. In: Proc. of 4th World Enformatica Congress, International Conference on Pattern Recognition and Computer Vision, Istanbul, Turkey (June 24 - 26 2005) 213–217
11. Einsele, F., Ingold, R., Hennebert, J.: Using hmms to recognize ultra low resolution anti-aliased words. In: Technical report 283-07-05, University of Fribourg, Switzerland. (2007)
12. Nagy, G.: Twenty years of document image analysis in pami. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (Jan 2000) 38–62
13. Einsele, F., Hennebert, J., Ingold, R.: Towards identification of very low resolution, anti-aliased characters. In: Proc. of ISSPA07, Sharjah, UAE (Feb 12 - 17 2007)

---

# Manageable Phrase-Based Statistical Machine Translation Models

Ghada Badr, Eric Joanis, Samuel Larkin, and Roland Kuhn

Institute for Information Technology National Research Council of Canada, NRC  
{GhadaHany.Badr, Eric.Joanis, Samuel.Larkin, Roland.Kuhn}@nrc.gc.ca

**Summary.** Statistical Machine Translation (SMT) is an evolving field where many techniques in Syntactic Pattern Recognition (SPR) are needed and applied. A typical phrase-based SMT system for translating from a  $T$  (target) language to an  $S$  (source) language contains one or more  $n$ -gram language models (LMs) and one or more phrase translation models (TMs). These LMs and TMs have a large memory footprint (up to several gigabytes). This paper describes novel techniques for filtering these models that ensure only relevant patterns in the LMs and TMs are loaded during translation. In experiments on a large Chinese-English task, these techniques yielded significant reductions in the amount of information loaded during translation: up to 58% reduction for LMs, and up to 75% for TMs.

## 1 Introduction

### 1.1 Background

This paper focuses on efficient filtering techniques for the components that take up the largest part of SMT system memory: the LMs and TMs. Though the paper's experimental results pertain to a particular SMT system, this system is typical of a wide range of SMT systems, such as [7, 8, 10, 12, 13]. Each LM component of an SMT system returns the probability of a word sequence in the  $T$ -language; we use  $n$ -gram LMs, which estimate the conditional probability that the  $i$ th word follows the preceding  $i - 1$  words [5]. We often employ several different LMs, each trained on a unilingual  $T$  corpus, and several TMs, each trained on a bilingual corpus. Each TM is stored in a "phrase table": a list of triples  $(s, t, m)$  where  $s$  is a contiguous sequence of  $S$  words,  $t$  is a contiguous sequence of  $T$  words, and  $m$  measures the strength of the association between them. These triples are extracted from a parallel bilingual corpus using the "DiagAnd" method described in [9] with IBM 2 alignment. Each LM and TM receives a loglinear weight optimized by the algorithm in [11]. During decoding, the decoder matches subsequences of the input  $S$ -language sentence in the TMs. A translation hypothesis is formed by concatenating  $T$  phrases from such matching phrase pairs [8]. Every hypothesis generated by the system for

an input sentence contains only words that appear in at least one  $t$  such that  $(s,t)$  is a phrase pair in a TM. Thus, given the input sentence and the TMs, one knows which  $T$ -language words in the system's vocabulary cannot possibly appear in the translation hypotheses. TMs and TMs are usually represented as *tries* [6] in memory.

Work on minimizing SMT memory requirements includes pruning of TMs for decoding based on a rudimentary analysis of the input sentences [8]; here, we show how to further prune the set of loaded phrase pairs by considering the relationship between different TMs. The work most closely related to ours for LM pruning involves a bag-of-words algorithm for the input  $S$  document [7, 10, 13]. The algorithm, "Document-Vocabulary LM Filtering" (*Doc-Voc-LM*), puts into a "bag" all and only the  $T$  words that could be used for translation, based on all sentences in the document and the phrase pairs in the TMs; an  $n$ -gram will only be loaded if all the words in it are in this "bag". A related technique has been applied to SMT rescoring [7].

This paper introduces three new techniques for "batch mode" translation of a document: "Sentence-Vocabulary LM Filtering" (*Sent-Voc-LM*) and "Sentence-Phrase LM Filtering" (*Sent-Phrase-LM*) for filtering LMs, and "Limit-TM" for filtering TMs. In the standard *Doc-Voc-LM* filtering approach, an  $n$ -gram from the LM will be loaded if all words in it come from the bag of target words for the input document. By contrast, *Sent-Voc-LM* creates a separate bag of words for each sentence in the document, and  $n$ -grams are only loaded if all words in them are in the current sentence's bag. *Sent-Phrase-LM* goes further, by prohibiting  $n$ -grams that cannot be generated from the phrases associated with the current sentence - e.g., if  $xyz$  is the only phrase with target word  $y$  associated with this sentence, then even if word  $w$  occurs in other phrases associated with the sentence, *Sent-Phrase-LM* will not load  $n$ -grams  $wy$  and  $yw$ . Initial experiments with *Sent-Voc-LM* and *Sent-Phrase-LM* yielded only a small reduction in the number of  $n$ -grams loaded from the LM. However, study of the relationship between multiple TMs used by the system yielded *Limit-TM*, which finds a much smaller set of phrases associated with each input sentence (without loss of information). Since this results in a smaller bag of target words per sentence, *Sent-Voc-LM* and *Sent-Phrase-LM* performed very well when *Limit-TM* was applied. Thus, our system can now load larger models and therefore generate better translations.

## 2 Language Model Filtering

We now explain *Sent-Voc-LM* and *Sent-Phrase-LM* in more detail.

### 2.1 Sent-Voc-LM

As mentioned above, one way of filtering the LM is to apply *Doc-Voc-LM*. However, as the number of  $S$  sentences in the input document increases, the size of the bag of words increases, and the ability to filter LMs decreases. One could

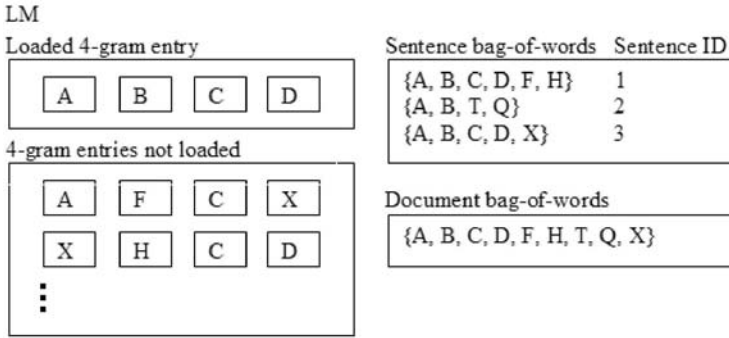


Fig. 1. Example of  $n$ -grams loaded by Doc-Voc-LM but not by Sent-Voc-LM

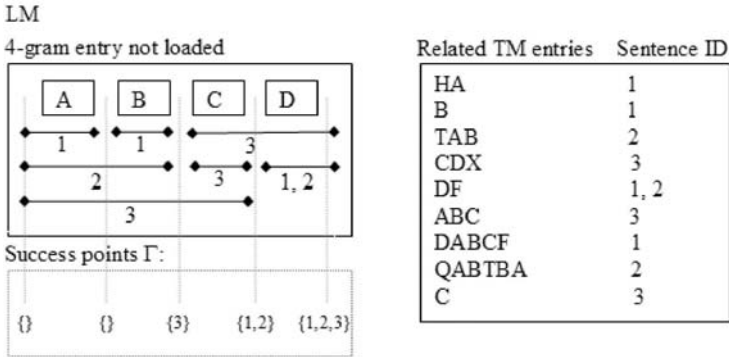


Fig. 2. Example of an  $n$ -gram loaded by Sent-Voc-LM but not by Sent-Phrase-LM

load separate LMs for each  $S$ -sentence, but that would take too long; it is better to load LMs once for a group of sentences.

The *Sent-Voc-LM* filter uses per-sentence bags of words: each bag contains only the  $T$  words that might be needed to translate one  $S$  sentence in the input<sup>1</sup>. Only  $n$ -grams that consist of words that are all found together in at least one bag are loaded. For unigrams, this is no different from *Doc-Voc-LM*, but for bigrams and higher order  $n$ -grams, it makes a big difference. *E.g.*, we no longer load a bigram  $(x, y)$  when  $x$  is found only in the bag for sentence  $i$ , and  $y$  is only in the bag for a different sentence  $j$ . Figure 1 shows how *Sent-Voc-LM* can filter out  $n$ -grams that would be loaded by *Doc-Voc-LM*. Here, 4-grams “AFCX” and “XHCD” will be allowed by the latter (since they are made of words from the document bag-of-words) but not by the former (since they do not come from a per-sentence bag-of-words); “ABCD” will be loaded by both.

<sup>1</sup> The out-of-vocabulary words in  $S$  are part of each per-sentence bag of words to ensure that the corresponding  $n$ -gram entries will be loaded.

## 2.2 Sent-Phrase-LM

(*Sent-Phrase-LM*) filters out even more out  $n$ -grams from the LM by prohibiting those that cannot be generated by joining phrases from the TM (even when all the words in the  $n$ -gram are in the same per-sentence bag of words). In Figure 2, the  $n$ -gram “ABCD” would be loaded by *Doc-Voc-LM* and *Sent-Voc-LM*, but is filtered out by *Sent-Phrase-LM* because it cannot be generated by combining phrases for the same sentence.

There are three ways that an  $n$ -gram can appear in a translation for a given sentence: the  $n$ -gram appears inside a phrase, the  $n$ -gram is formed by a suffix<sup>2</sup> of one phrase and a prefix of the phrase that follows it in the translation, or the  $n$ -gram is formed by a suffix of one phrase, then one or more complete phrases, and then a prefix of the following phrase. To implement (*Sent-Phrase-LM* efficiently, we used suffix-trees [1, 4]. The suffix-tree stores all possible  $T$  phrases including pre-/in-/suffixes of these phrases. Each node in the tree is marked to indicate whether it is a complete phrase, a prefix, a suffix, or an infix of a phrase.

We implemented the *Sent-Phrase-LM* criterion using dynamic programming [3]. The algorithm uses a  $T$  vector to tabulate the success points in each  $n$ -gram that can be generated, and uses the suffix-tree to retrieve the complete and pre-/in-/suffix sets. The algorithm requires at most  $O(n)$  lookups in the suffix-tree to generate all success point sets.  $n$  is usually small (3 – 6), so it requires only a few lookups and some set operations to decide if an  $n$ -gram should be kept. For details, see [2].

## 3 Translation Model Filtering

Phrase	TM <sub>1</sub> score	TM <sub>2</sub> score	<i>Limit-TM</i> decision
<i>a</i>	-1	-1	keep
<i>b</i>	-2	-10	keep
<i>c</i>	-3	-3	keep
<i>d</i>	-10	-2	keep
<i>e</i>	-2.5	-9	keep
<i>f</i>	-3.5	-4	reject

**Fig. 3.** *Limit-TM* Example with two TMs

tences in a *trie* [6] data structure. We call this technique, already used by most SMT systems, *Grep-TM*.

### 3.1 *Limit-TM*

*Limit-TM* goes beyond *Grep-TM* by considering relationships between phrase pairs in different TMs. In a TM, we may have many  $T$  phrases for a given  $S$

<sup>2</sup> We assume that a phrase is a prefix, a suffix, and an infix of itself.

There are two reasons for filtering TMs: to reduce the memory taken up by the TMs themselves, and to reduce the vocabulary to make LM filtering more effective. In this section, we propose an efficient algorithm for filtering TMs. We begin with a reasonable baseline that keeps only phrase pairs whose  $S$ -language phrases occur in at least one of the sentences to be translated. This can be done efficiently by storing all  $S$ -language phrases from the source sentences

We call this technique, already used by most

phrase. During decoding, most SMT systems limit the number of  $T$  phrases, considering only the  $L$  best ones for a given  $S$  phrase. With only one TM, for each  $S$ -language phrase, one thus keeps only the  $L$   $T$  phrases with the highest score. However, one usually has more than one TM. *Limit-TM* chooses the  $L$   $T$ -language phrases to be kept by analyzing their scores in all TMs and the weights with which they may be combined.

When optimizing decoder weights, we decode the same sentences many times with different weights. Therefore, we want *Limit-TM* to filter TMs ahead of time in a way that will be correct regardless of the TM weights (for fixed  $L$ ). Assume the weights are non-negative. Given a  $T$  phrase,  $tp_i$ , if there are  $L$  other  $T$  phrases that are “better” than  $tp_i$  according to all TMs, then no matter what the weights,  $tp_i$  will remain behind those  $L$  other  $T$  phrases and will never be used during decoding. In this case,  $tp_i$  could be filtered out from all TMs. For example, in Table 3 where we have two TMs and  $L = 2$ ,  $a$  will have the top score among those phrases. Which phrase comes second, however, depends on the weights: it will be  $b$  in  $TM_1$ ,  $c$ , or  $d$  in  $TM_2$ , and therefore it must always be kept. But  $e$  will never be among the top two phrases:  $a$  will always have the highest score, and then one of  $b$  or  $c$  will always be better than  $e$ . However,  $e$  will not be filtered by our criterion because detecting that it can be requires expensive comparisons involving multiple phrase table entries. Only  $f$  can be filtered out, since  $a$  and  $c$  are better than  $f$  in both tables; typically, many more entries like  $f$  can be filtered out. This example shows that *Limit-TM* is sufficient but not necessary. The proof can be found in [2].

*Limit-TM* can be implemented efficiently using dynamic programming [3]. With  $L$  defined as before, let  $n$  be the number of TMs,  $I$  be the number of unique target phrases across all TMs for a given  $S$ -language phrase,  $tp_i$  be a  $T$ -language phrase. In the worst case the algorithm can take up to  $O(I \log(I))$  for sorting plus  $O(I^2 N)$  for *Limit-TM* itself, but on average  $O(ILN)$ . Empirically, we found that this algorithm runs reasonably fast even for large values of  $I$ . For details, see [2].

## 4 Experimental Results

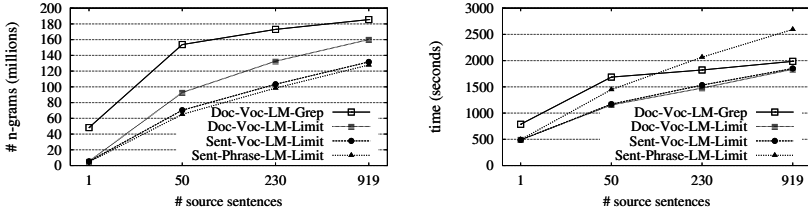
### 4.1 Data Set and Models

These techniques were applied to our Chinese-English SMT system. The TM training corpora were those distributed for the NIST MT06 Chinese evaluation, large-data track (<http://www.nist.gov/speech/tests/mt>). For LMs, we additionally used the English Gigaword corpus (LDC2005T12). The test corpus consists of the 919 variable-length Chinese sentences from the Multiple Translation Chinese Corpus, Part 4 (LDC2006T04).

We tested the proposed LM and TM filters using the two different LMs and five different TMs included in our highest-scoring NIST06 Chinese-English system. There were two training corpora for LMs: one has 3.2M sentences and the other (Gigaword) has 13M Sentences. For each LM training corpus, we generated 3-, 4-, and 5-gram models. For instance, the Giga corpus yields a 4-gram LM

**Table 1.** TM sizes in MBs, total time in minutes (loading, filtering and writing), total vocabulary size (919 sentences), and average vocabulary size per sentence.  $L = 30$ .

	Multi-Prob-TM	Grep-TM	Limit-TM
TM	958M	122M	29M
Time	57m	14m	68m
Total vocabulary	1,783,641	268,375	69,644
Average vocabulary	-	11,891	1,581

**Fig. 4.**  $n$ -gram counts (left) and total loading times (right) for *Giga-4g-LM* as function of no. of source Chinese sentences for different techniques

of size 2,000MB and a 5-gram LM of size 3,200MB. We trained the five TMs on a total of 8.2M sentences. We then combined all of them in one table with multiple probabilities: *Multi-Prob-TM* (958MB). In this table, each phrase pair is associated with its probabilities from various TMs. Phrases missing from a TM are given a small, arbitrary probability value.

## 4.2 Experimental Setup and Results

As a baseline for TM filtering, the *Multi-Prob-TM* was filtered using the *Grep-TM* technique. We then applied the *Limit-TM* technique to *Grep-TM* output. For each LM, we created two baselines using *Doc-Voc-LM*. The first baseline, *Doc-Voc-LM-Grep*, was created by using the vocabulary collected from *Grep-TM*, and the second one, *Doc-Voc-LM-Limit*, was created using vocabulary collected from *Limit-TM*.

Experiments were conducted to compare *Sent-Voc-LM* and *Sent-Phrase-LM* with *Doc-Voc-LM-Grep*, in the case where the only TM filtering is *Grep-TM*. Results did not show a big reduction in LM sizes in this case. These disappointing results were not unexpected: Table 1 shows that the average size of vocabulary per sentence was 11,891! Note (below) that we get much better results when combining LM filtering with *Limit-TM*. In fact, these LM filtering techniques can be effectively combined with any TM filtering technique that shrinks average per-sentence vocabulary.

Our TM filtering experiments compared *Limit-TM* with *Grep-TM*: Table 1 shows that *Limit-TM* reduces the combined TM size by more than 75% when the



limit,  $L$ , is set to  $30^3$ . The table also shows that *Limit-TM* reduces the average per-sentence vocabulary by nearly an order of magnitude, greatly improving the effectiveness of LM filtering. The table also shows processing time.

Finally, we carried out experiments to measure the impact of LM filtering for the two kinds of TM filtering. Figure 4 (left) shows the number of  $n$ -grams in millions after applying the techniques on *Giga-4g-LM*, as a function of number of sentences. The result shown here is typical of most  $n$ -gram LMs. Except for the “Doc-Voc-LM-Grep” line, results were based on TMs that were filtered using *Limit-TM*. The top two lines show that *Limit-TM* significantly reduces LM requirements (*Doc-Voc-LM-Limit* uses up to 40% fewer 4-grams than *Doc-Voc-LM-Grep*). The figure also shows that *Sent-Voc-LM-Limit* performs much better with *Limit-TM*. For example, for Gigaword 4-gram LM and 50 source sentences, *Sent-Voc-LM-Limit* uses about 54% fewer 4-grams than *Doc-Voc-LM-Grep*, and about 24% fewer than *Doc-Voc-LM-Limit*. Under the same conditions, *Sent-Phrase-LM-Limit*<sup>4</sup> uses about 58% fewer 4-grams than *Doc-Voc-LM-Grep*, and about 30% less 4-grams than *Doc-Voc-LM-Limit*. The time requirement is shown in Figure 4 (right). *Sent-Voc-LM-Limit* combines effective filtering and speed. It can be done on the fly while loading models for decoding, and slightly speeds up decoding (evidence not shown here).

## 5 Conclusions

In this paper, we described three techniques for reducing the size of SMT models in memory by loading only entries that are needed for translation. This will make it possible to use larger models, yielding improved translations. Two LM filtering techniques, *Sent-Voc-LM* and *Sent-Phrase-LM*, were proposed; they work best with a small per-sentence vocabulary. A TM filtering technique, *Limit-TM*, was presented to reduce the size of the TMs. *Limit-TM* greatly reduces the  $T$ -language vocabulary size, which in turn improves LM filtering. The combined techniques showed reductions in LM sizes by up to 58%. *Limit-TM* showed a remarkable reduction in the joint TM size by up to 75%. These techniques enabled us for the first time to find room in memory for 5-gram LMs, which measurably improved the quality of the system’s output.

## References

1. A. Andersson and S. Nilsson. Efficient implementation of suffix trees. *SOFTPREX: Software-Practice and Experience*, 25(2):129–141, 1995.
2. G. Badr, E. Joanis, and S. Larkin. Manageable phrase-based statistical machine translation models. Technical report nrc 49304, 2007.
3. T. H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 1990.

<sup>3</sup> We tested our decoder with several values for  $L$ ;  $L = 30$  was optimal.

<sup>4</sup> *Sent-Phrase-LM* performance strongly depends on the training data - it should be tried on other data.

4. M. Fuketa, T. Sumitomo, M. Shishibori, and J. Aoe. A suffix compression algorithm of tries. *ICCPOL'99: 18th International Conference on Computer Processing of Original Languages*, 18:345–348, 1999.
5. J. T. Goodman. A bit of progress in language modeling. Technical Report MSR-TR-2001-72, 2001.
6. P. Jacquet and W. Szpankowski. Analysis of digital tries with Markovian dependency. *IEEE Trans. Information Theory, IT-*, 37(5):1470–1475, 1991.
7. K. Kirchhoff, K. Duh, and C. Lim. The University of Washington machine translation system for IWSLT 2006. In *Proc. of the International Workshop on Spoken Language Translation*, pages 145–152, Kyoto, Japan, 2006.
8. P. Koehn. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. Washington DC, 2004.
9. P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conf. (HLT-NAACL)*, pages 127–133, Edmonton, Canada, June 2003.
10. A. Mauser, R. Zens, E. Matusov, S. Hasan, and H. Ney. The RWTH statistical machine translation system for the IWSLT 2006 evaluation. In *Proc. of the International Workshop on Spoken Language Translation*, pages 103–110, Kyoto, Japan, November 2006.
11. F. J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceeding of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, PA, July 2002.
12. F. Sadat, H. Johnson, A. Agbago, G. Foster, R. Kuhn, J. Martin, and A. Tikui-sis. Portage: A phrase-based machine translation system. In *The Association for Computational Linguistics (ACL) 2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages 133–136, Ann Arbor, Michigan, USA, 2005.
13. Wade Shen, Richard Zens, Nicola Bertoldi, and Marcello Federico. The JHU workshop 2006 IWSLT system. In *Proc. of the International Workshop on Spoken Language Translation*, pages 59–63, Kyoto, Japan, 2006.

---

# Automatic Detection of Disorders in a Continuous Speech with the Hidden Markov Models Approach

Marek Wiśniewski, Wiesława Kuniszyk-Józkowiak, Elżbieta Smołka,  
and Waldemar Suszyński

Institute of Computer Science, Maria Curie-Skłodowska University, Pl. Marii  
Curie-Skłodowskiej 1, 20-031 Lublin, Poland  
`marek.wisniewski@umcs.lublin.pl`

**Summary.** Hidden Markov Models are widely used for recognition of any patterns appearing in an input signal. In the work HMM's were used to recognize two kind of speech disorders in an acoustic signal: prolongation of fricative phonemes and blockades with repetition of stop phonemes.

In the work a tests results of a recognition effectiveness are presented for considered speech disorders by HMM models in different configurations. There were summary models applied for a class of disorder recognition, as well as models related to disturbance of individual phoneme. The tests were carried out by use of the author's implementation of HMM procedures.

## 1 Introduction

The classification of speech disorders involves many types of disturbances. Proper recognition of these has a very important significance for the choice of a therapy process [1, 2]. The equally essential problem is an objective evaluation of the kind of disorder as well as release a therapist from arduous rehearing and analyzing recorded utterances of stuttering people. So a further search for more accurate methods of automatic disturbance detection is desirable.

The HMMs are stochastic models that are widely used for recognition of various patterns appearing in an input signal. HMMs are used for description of a system state. However the state cannot be explicitly determined because it is hidden. Only observations generated by the model are given, and it is only the base on that one can estimate probability of being a system in a particular state. In the case of speech recognition systems an observation is an acoustic signal and the state is the recognized pattern (i.e. disfluency)[3].

In the recognition process with the HMM's there is a creation of the database of models required. Every model is designed for recognition of particular pattern (disfluency) appearing in a signal. Next there is a probability of emission of the analyzed fragment counted for every model in the database. The model that gives the greatest probability is then selected and if the probability is above the chosen threshold the recognition is done.

Every recognition model is prepared by training. Having a base HMM model  $\lambda = (\pi, A, B)$  and a sufficient number of samples of the same pattern, one can prepare a model, so that it achieves maximum emission probability for that pattern.

For requirements of this paper there were several models learned designed to recognition of prolonged fricative phonemes and blockades with repetition of stop phonemes. That disturbances are the most often presented in nonfluent speech.

## 2 Sample Parameterization

The acoustic signal requires to be parameterized before analysis. The most often used set of parameters in the case are Mel Frequency Cepstral Coefficients (MFCC). The process of determining MFCC parameters in the work is as follows:

- splitting signals into frames of 512 samples' length,
- FFT (Fast Fourier Transform) analysis on every frame,
- transition from linear to mel frequency scale according to the formula:  $F_{mel} = 2595 \log(1 + F/700)$  [4, 5],
- signal frequency filtering by 20 triangular filters,
- calculation of the required (20) number of MFCC parameters.

The elements of each filter are determined by summing up the convolution results of the power spectrum with a given filter amplitude, according to the formula:

$$S_k = \sum_{j=0}^J P_j A_{k,j},$$

where:  $S_k$ - power spectrum coefficient,  $J$ - subsequent frequency ranges from FFT analysis,  $P_j$ - average power of an input signal for  $j$  frequency,  $A_{k,j}$ -  $k$ -filter coefficient.

With  $S_k$  values for each filter given, cepstrum parameter in the mel scale can be determined [6]:

$$MFCC_k = \sum_{k=1}^K (\log S_k) \cos \left[ n(k - 0.5) \frac{\pi}{K} \right], \text{ for } n = 1..N,$$

where:  $N$  - required number of MFCC parameters,  $S_k$ - power spectrum coefficients,  $K$ - number of filters.

The justification of the transition from the linear scale to mel scale is that the latter reflects the human perception of sounds better.

Division of an audio sample into frames leads to disturbance of real signal parameters. For avoidance of this problem each value of an audio frame were multiplied by proper coefficient of the Hamming window [7], according to the formula:

$$W[n] = 0.54 - 0.56 \cos \left( \frac{2\pi k}{n - 1} \right),$$

where:  $n$ - sample number,  $N$ - frame width.

However use of that formula cause losing of the information on a frame boundaries. That's way there were partial overlapping of successive frames applied. The length of an overlapp was adjusted to 1/3 of the frame length (rounded to 170 points).

### 3 Codebook Preparation

The MFCC analysis of the acoustic signal gives too many parameters to be analyzed with the application of the HMM with a discrete output. At the same time, the number of MFCC parameters cannot be decreased, since then important information may be lost and so the effectiveness of recognition may be poor.

In order to reduce the number of parameters, encoding with a proper codebook can be applied [7]. Preparation of the codebook is as follows. First, the proper sample of an utterance needs to be chosen, which covers the entire acoustic space to be examined. Next it can be generated, for example by the use the „k-means“ algorithm. Three fragments of utterances were selected, each lasting 54 seconds and articulated by three different persons and, afterwards MFCC coefficients were calculated. The obtained set of parameters were divided into appropriate number of regions and their centroids were found. For counting the distances between vectors the Euclidean formula were used:

$$d_{x,y} = \sqrt{\sum_{i=1}^N (x_i - y_j)^2}$$

where:  $d_{x,y}$ - the Euclidean distance between  $N$ -dimensional vectors  $X$  and  $Y$ .

For the examination there is to determine a size of the codebook. The size should be such so a recognition ratio is on an acceptable level and a computation time is reasonable. Based on own studies there were a codebook prepared with 512 elements and used for testing.

### 4 Testing Procedure

For tests purpose there was an audio sample of a length of 87624 ms prepared. The sample contained 24 disfluencies (10 stops blockades with repetitions of and 14 prolongations of fricatives). Every kind of disturbance appeared two times in the sample and they were: C, s, z, x, Z, v, S (fricatives) and p, t, k, b, g (stops). For every kind of disfluency there were suitable model prepared and additionally two summary models for a disturbance class recognition (stop blockades and fricatives prolongation). During the tests it has appeared that, in the case of stop blockades, a silence have negative influence on recognition ratio. To eliminate of this an additional groups of models for stop blockades were trained with the use of samples freed from silence.

Base models had 8 states and 512 code symbols. Probability values for matrixes A, B,  $\pi$  were randomly generated. From 3 to 6 patterns of the same

disfluency were utilized for training every model. In the case of summary models the number of patterns was much greater: for prolongation of fricatives recognition model there was 38 samples used, for stops blockade of recognition model - 30 samples and for summary model learned with patterns that were free of silence - 30 samples.

For testing, the HMM application was used, where appropriate algorithms were implemented. Parameters of the sound samples which were used were as follows: sample frequency: 22050Hz, amplitude resolution: 16 bits. All the recordings were normalized to the same dynamic range and encoded with the use of earlier prepared codebook.

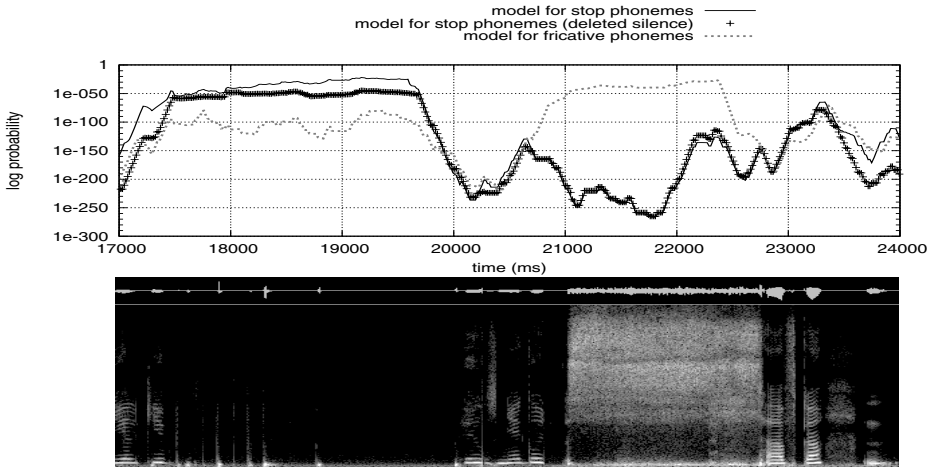
The examination of recognition effectiveness was carried on in the following way. From the sample, segments of the proper length were taken (30 code symbol - 465 ms and 60 code symbols - 930 ms) with the step of 1 symbol (about 15.51 ms) and then an emission probabilities for each model were counted.

From the obtained results there were distribution of emission probabilities across the time graph prepared for every model. Next, in the experimental way, the threshold value of a probability was chosen. A fragment that achieved the probability greater than the threshold was considered as disfluency. After cutting off lower probabilities from a graph there were only fragments that indicated disfluency. Fragments appearance time (read out from a graph) were compared with the time read out from a spectrogram. If both the times were equal and a disturbance was indicated by a proper model then it was considered as correct recognition.

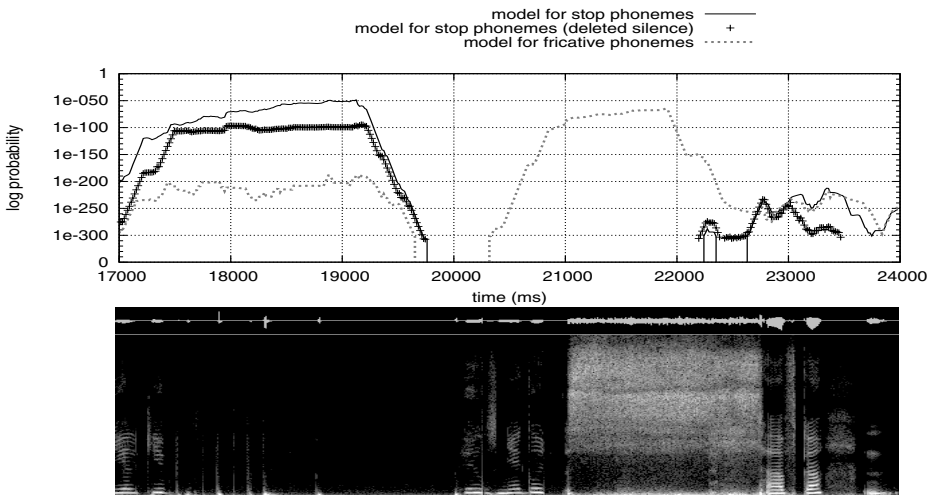
In the figure 1 and 2 there are examples of probability distribution shown for three summary models and an utterance spectrogram[8]. Obtained probabilities cover a very big range so the logarithmic scale was used. The place where a disorder appears is characterized by a very high probability value in comparison to other places. For example the probability difference between two disorders that appears (stop and fricative at a time range 17500-19500 ms) is over  $1E-20$ , so it indicates a stop phoneme blockade.

The value of probability depends on a window size - when the window is longer the probability is lower. One can notice that in the case of the 60 frames window length the graph has smaller fluctuations, but on the other side the 30 frames window length graph is more detailed.

There were two groups of test carried on in the work. In the first case (tables 1-4) there were used summary models for fricatives, summary models for stops as well as models learned for every individual phoneme separately. In the second case (tables 5-8), for stop phonemes there were applied models trained with samples with deleted silence (models for fricatives were the same). One should notice that models learned for individual phonemes was used for recognition of a class of disfluency (like summary models) and not for recognition of individual phoneme disturbances. Speech disorders are often sounds that are completely different form known phonemes and it is almost impossible to recognize nonfluency of individual phoneme. There are test results of recognition effectiveness shown below for the second groups of models (learned by samples with deleted silence).



**Fig. 1.** The analysis result of the utterance: "vjelci bbb b1w z ego ctSwovjek ssss safka" for the window of 30 frames length (465 ms); probability distribution for three 8-state models with the codebook size of 512-elements (top); the spectrogram (bottom)



**Fig. 2.** The analysis result of the utterance: "vjelci bbb b1w z ego ctSwovjek ssss safka" for the window of 60 frames length (930 ms); probability distribution for three 8-state models with the codebook size of 512-elements(top); spectrogram (bottom)

For the comparison of recognition ratio two parameters are useful: sensitivity and predictability. They were counted according to formulas [9]:

$$\text{sensitivity} = ((\text{number of correctly recognized nonfluencies}) / (\text{number of all nonfluencies in the sample})) * 100\%;$$

**Table 1.** The recognition ratio for summary models; window size 30 frames, probability threshold 1E-45

kind of disturbance	false recognitions	lack of recognitions	correct recognitions	predictability	sensitivity	correctness
stops blockades with repetition	7	2	8	47%	80%	27%
prolongation of fricatives	0	9	5	100%	36%	36%
summary:	7	11	13	65%	54%	19%

**Table 2.** The recognition ratio for summary models; window size 60 frames, probability threshold 1E-90

kind of disturbance	false recognitions	lack of recognitions	correct recognitions	predictability	sensitivity	correctness
stops blockades with repetition	0	2	8	100%	80%	80%
prolongation of fricatives	0	12	2	100%	14%	14%
summary:	0	14	10	100%	42%	42%

**Table 3.** The recognition ratio for models prepared for individual nonfluent phonemes; window size 30 frames, probability threshold 1E-45

kind of disturbance	false recognitions	lack of recognitions	correct recognitions	predictability	sensitivity	correctness
stops blockades with repetition	2	4	6	75%	60%	35%
prolongation of fricatives	2	8	6	75%	43%	18%
summary:	4	12	12	75%	50%	25%

$$\text{predictability} = \left( \frac{\text{number of correctly recognized nonfluencies}}{\text{correctly recognized nonfluencies} + \text{number of false nonfluency recognition}} \right) * 100\%;$$

As a more general parameter the following formula was used:

$$\text{correctnes} = \text{sensitivity} * (100\% - \text{predictability})$$

If the above formula was giving a negative value then this value was treated as null percentage.



**Table 4.** The recognition ratio for models prepared for individual nonfluent phonemes; window size 60 frames, probability threshold 1E-60

kind of disturbance	false recognitions	lack of recognitions	correct recognitions	predictability	sensitivity	correctness
stops blockades with repetition	0	2	8	100%	80%	80%
prolongation of fricatives	0	11	3	100%	21%	21%
summary:	0	13	11	100%	45%	45%

**Table 5.** The recognition ratio for summary models with „a deleted silence“; window size 30 frames, probability threshold 1E-49

kind of disturbance	false recognitions	lack of recognitions	correct recognitions	predictability	sensitivity	correctness
stops blockades with repetition	2	5	5	71%	50%	21%
prolongation of fricatives	0	8	6	100%	43%	43%
summary:	2	13	11	85%	45%	30%

**Table 6.** The recognition ratio for summary models with „a deleted silence“; window size 60 frames, probability threshold 1E-145

kind of disturbance	false recognitions	lack of recognitions	correct recognitions	predictability	sensitivity	correctness
stops blockades with repetition	0	2	8	100%	80%	80%
prolongation of fricatives	1	4	10	91%	71%	62%
summary:	1	6	18	95%	75%	70%

According to the results, the best recognition ratio was achieved for summary models with deleted silence (table 6) and for the window length of 60 frames (correctness equal to 70%). Such value is satisfactory. In general, the correctness was better when the window size was equal to 60 frames (only in one case was differently) and for models with deleted silence (also only in one case was differently). The predictability coefficient was always better in every test when 60 frames window was used, so there were less incorrect recognitions.

**Table 7.** The recognition ratio for models prepared for individual nonfluent phonemes and with „a deleted silence“; window size 30 frames, probability threshold 1E-49

kind of disturbance	false recognitions	lack of recognitions	correct recognitions	predictability	sensitivity	correctness
stops blockades with repetition	3	6	4	57%	40%	0%
prolongation of fricatives	3	6	8	72%	57%	29%
summary:	6	12	12	67%	50%	27%

**Table 8.** The recognition ratio for models prepared for individual nonfluent phonemes and with „a deleted silence“; window size 60 frames, probability threshold 1E-145

kind of disturbance	false recognitions	lack of recognitions	correct recognitions	predictability	sensitivity	correctness
stops blockades with repetition	1	4	6	86%	60%	46%
prolongation of fricatives	6	9	5	45%	36%	0%
summary:	7	13	11	67%	45%	12%

## 5 Summary

The recognition ratio of speech disorders with the use of HMM mainly depends on a very accurate selection of teaching patterns. In the case of blockades with repetitions the problem is a silence that appears in a sample. Sometime it leads to incorrect interpretation - silence is recognized as a disturbance. An another important conclusion is, that a proper selection of window length is crucial. The longer sections cause that there is less fluctuations and the number of incorrect recognitions is also lesser. On the other side a selection of too wide window leads to lesser sensitivity.

In the context of a recognition process very important is a selection of a proper probability threshold. It is a compromise between the sensitivity and predictability level. In the future authors plan to implement procedures for automatic selection of that threshold.

## References

- [1] Kuniszyk-Jóźkowiak W., Smółka E., Suszyński W.: Akustyczna analiza niepłynności w wypowiedziach osób jękaających się, *Technologia mowy i języka*. Poznań 2001
- [2] Suszyński W.: *Komputerowa analiza i rozpoznawanie niepłynności mowy*, rozprawa doktorska, Gliwice 2005

- [3] Deller J. R., Hansen J. H. L., Proakis J. G.: Discrete-Time Processing of Speech Signals, IEEE, New York 2000
- [4] Wahab A., See Ng G., Dickiyanto, R.: Speaker Verification System Based on Human Auditory and Fuzzy Neural Network System, Neurocomputing Manuscript Draft, Singapore
- [5] Picone J.W.: Signal modeling techniques in speech recognition, Proceedings of the IEEE, 1993, 81(9): 1215-1247
- [6] Schroeder, M.R.: Recognition of complex acoustic signals, Life Science Research Report, T.H. Bullock, Ed., (Abakon Verlag, Berlin) vol. 55, pp. 323-328, 1977
- [7] Tadeusiewicz R.: Sygnał mowy, Warszawa 1988
- [8] Horne R. S.: Spectrogram for Windows, ver. 3.2.1
- [9] Barro S., Marin R., Fuzzy logic in medicine, Phisica-Verlag, A Springer-Verlag Company, Heidelberg, New York, 2002

---

# The Polish Finger Alphabet Hand Postures Recognition Using Elastic Graph Matching

Joanna Marnik

Rzeszow University of Technology, ul. W. Pola 2, 35-235 Rzeszow  
jmarnik@prz-rzeszow.pl

**Summary.** In the present paper the results of applying the Elastic Graph Matching (EGM) method to classify the 23 postures occurring in Polish finger alphabet are presented. Four different classifiers were constructed. Three of them used different feature vectors to the graph's nodes description. The fourth one integrated the former three classifiers.

## 1 Introduction

Fast expansion in the human-computer interaction domain has occurred in the last years. Hand gestures recognition is an important task in this area. Automatically recognized hand gestures can be used in the human-computer interaction to issue instructions in a more natural way. Also communication with robots working in the environment hostile or noisy for humans is possible by means of hand gestures. Hands are the most functional part of the body and this is why they are used to convey some information, e.g. in deaf community.

This work concerns postures occurring in Polish Finger Alphabet (PFA). Gestures of PFA are used in Polish sign language to convey names and abbreviations, to spell words which do not have ideographic signs and to convey formants in derivative words and inflexional endings [4]. Set of the PFA gestures is shown on Fig. 1. Hand shapes occurring in finger alphabets were the subject of the Grobel and Hienz work [3]. In their system coloured gloves were used to facilitate fingers detection. 31 letters and 7 additional hand shapes used in German sign language were considered. Special lighting requirements were met. Rule based classification was performed. An average recognition rate of 93 % was achieved. The finger alphabet signs were the subject of Jiangqin and Wen's research [5], too. CyberGlove and Polhemus 3-D tracker were used to measure the position, movement and orientation of each finger and wrist. The multilayer neural network was used to recognize 30 gestures of the Chinese finger alphabet. The recognition rate was 98 %. The Polish finger alphabet signs were recognized by means of the morphological hit-or-miss transformation, orientation histograms and neural networks [7, 8]. The method has been tested on the image database of about 12000 static images obtained from three people. Recognition accuracy accounts to 87 %.

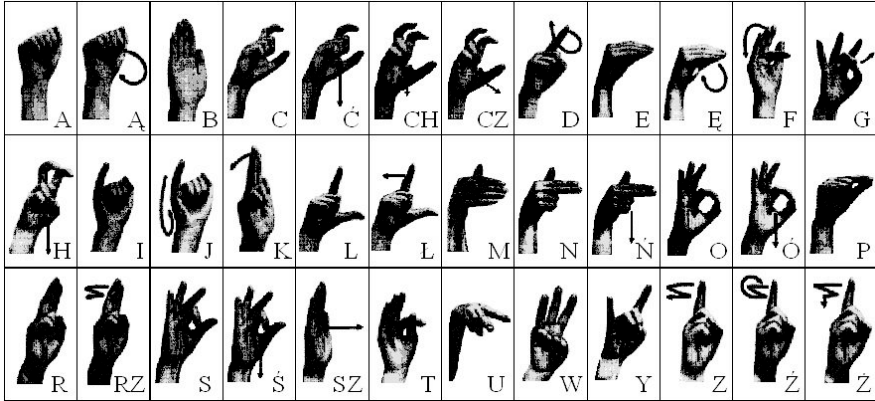


Fig. 1. Polish Finger Alphabet signs

The goal of the paper was to compute a recognition rate of the EGM [9, 10] method applied to 23 postures occurring in the PFA. The method was successfully applied to object recognition [6], to face finding and recognition [11], and also hand postures recognition [9, 10]. The number of postures considered in [9, 10] was significantly smaller (10 and 12, respectively) than in the present paper. Because a complex background was adopted in the papers quoted, features used in the recognition were obtained using Gabor filters for graylevel images [9, 10], as well as skin color information and Gabor filters, which were applied to images expressing each pixel's similarity to skin color [10]. In the present paper features based on Gabor filters were used, too. Additionally, features obtained from edge information and the distance to the camera were examined. The first kind of features could be used here, assuming that there are no objects at the background which have color similar to the human skin color. The second kind asserts information important for recognising the spatial postures, e.g. C, Ch, H. The EGM method and features used in the recognition process are presented in Sect. 2. A separate classifier was assigned to each kind of features. One hybrid classifier was built on the basis of all the three classifiers. Each of the four classifiers was tested using the image database created mainly for the purpose of the present paper. The data base contains hand postures occurring in the PFA. The four classifiers as well as the experiments carried out with the use of the classifiers and the image database are described in Sect. 3. Sect. 4 contains conclusions and a plan for further work.

## 2 Elastic Graphs

In the EGM method [9, 10] a shape is represented by a graph with a two-dimensional topology (Fig. 2). The graph nodes are labelled with a local image description called *jet*. A distance vector is assigned to every graph edge. A



**Fig. 2.** Elastic graph superimposed on the A posture model image

graph model is matched to the examined image. Firstly, a preliminary position of the graph is searched for on the basis of similarity function  $S^r$ . At this point transition and rotation transformations are taken into account.  $S^r$  function is calculated as the mean value of the similarities  $S^r(j)$  of the jets  $j$ ,  $j = 1, 2, \dots, P$  ( $P$  – number of the graph’s nodes,  $r$  – kind of jets).

$$S^r = \frac{1}{P} \sum_{j=1}^P S^r(j) \quad (1)$$

The jet’s similarity functions  $S^r(j)$  used in the present paper will be presented later. Secondly, each node of the graph is moved to a new position, the one where (a) the local image description attached to this node matches the image region around the position where the node is placed, according to the similarity function of the jets  $S^r(j)$ , and (b) the graph is not distorted too much. The latter requirement is expressed by the topology changes cost function, defined as:

$$C_{topol} = \frac{\left| \sum_{i=1}^K L_1(i) - \sum_{i=1}^K L_0(i) \right|}{K \cdot \sum_{i=1}^K L_0(i)} \quad (2)$$

where:  $K$  – number of edges,  $L_0(i)$  – length of  $i$ -th edge in the graph model,  $L_1(i)$  – length of  $i$ -th edge in the graph after matching. The final value of similarity of the model to the examined image  $S_c^r$  is calculated as:

$$S_{total}^r = S^r - \lambda \cdot C_{topol} \quad (3)$$

where  $\lambda$  is the costs coefficient.

The first jet’s similarity function is connected with Gabor jets, for creation of which Gabor wavelets were used [9, 10]. These wavelets represent a plane wave with the wave vector  $k$ , restricted by a Gaussian envelope function of the width  $\sigma$  [9, 10]. In the present paper Gabor jets were created by the use of 24 Gabor filters obtained for 8 different wave orientations, with 3 different wave frequencies each, and  $\sigma = 2.5$ , as in [9]. The similarity of such jets was calculated by the use of two functions proposed in [9]. The first of these functions  $S_{abs}^1$  uses only the complex filters responses magnitudes and is used in preliminary positioning of the graph. The second one  $S_{pha}^1$  also takes into account phases and is used in the

graph nodes repositioning. For the more similar jets bigger values of similarity function are obtained.

Jets of the second kind are composed of two values, obtained from the information about edges. The first value is determined as the maximum value  $M$  of the edge amplitudes occurring in the window  $\mathcal{O}$  surrounding a given node of the matched graph:  $M = \max_{(x,y) \in \mathcal{O}} | \text{grad}(x,y) |$ , where  $\text{grad}$  – image gradient. The second value is calculated as the value  $\alpha$  of the edge slope at the point for which the first value was found. For this kind of jets the similarity function is calculated as the average of the values  $S_M$  and  $S_\alpha$ , obtained for edge amplitudes and edge slopes, respectively:

$$S^2 = (S_M + S_\alpha) / 2. \quad (4)$$

$S_M$  describes similarity of the features associated with the maximum edge amplitudes  $M$  i  $M'$  for the model and the object, respectively. It is defined as  $S_M = | 1 - \frac{M'}{M} |$ .  $S_\alpha$  describes similarity of the features corresponding to edges orientations  $\alpha$  and  $\alpha'$  for the model and the object, respectively. It is calculated as the value of the acute angle between axes appointed by the directions  $\alpha$  i  $\alpha'$ . The resulting value was linearly scaled to the range  $[0, 1]$ .

The third kind of jets is obtained on the basis of the depth maps [1, 2]. The depth maps contain values of the distances between the corresponding points in the pair of rectified stereo images, named disparities. The disparity is inversely proportional to the distance to the camera. A jet was described by means of a single number, calculated as an average deviation  $D$  of the disparity  $d$  in the window  $\mathcal{O}$  from the average disparity  $D_{avg}$  corresponding to the hand area:  $D = \sum_{(x,y) \in \mathcal{O}} (d(x,y) - D_{avg}) / A$ ,  $A$  being the number of pixels in the window  $\mathcal{O}$ . The reference to the mean value enables the distance of the hand to the camera to change. Such jets similarity was defined as the normalized absolute value of the features  $D$  and  $D'$ , which correspond to the given node in the model and the object, respectively:

$$S^3 = |D - D'| / D_{max} \quad (5)$$

$D_{max}$  is the biggest possible disparity value. For the last two kinds of jets, smaller similarity values correspond to a better matching.

In the EGM method classification similarity values  $S_{total}^r$  (Eq. 3) are determined for each graph model after matching it to the examined object. The class of the object corresponds to the class's model which results in the best matching value. The value of the height to width ratio was assigned to each graph model. To accelerate the calculations, certain graphs were not considered during the classification, namely those for which the absolute value of the difference between the graph ratio and the examined hand shape ratio exceeded 0.4.

Before the classification an examined image was rotated so that the major axis of the hand was vertical. Next, it was scaled so that the hand height was the same as the height of the hand in the image for which the matched graph model was created. The model graph was placed so that centroid of the examined object is covered by the centroid of the object assigned to the graph model.

### 3 Experiments

Experiments were carried out on the image database consisting of 460 triples of test images presenting 23 PFA hand postures (Fig. 1), 20 images per posture, as well as on 70 triples of model images, 3–7 per posture. Each triple of images contained two gray-scale, rectified stereo images ( $320 \times 240$  pixels each) and one gray-scale image captured at the  $768 \times 576$  pixels resolution. The distance from the camera was not crucial. It was assumed that classifying posture B as Sz, G as O and D as Z and vice versa is not an error, because the elements of the pair are very similar.

Pairs-of-stereo-images based depth maps were obtained by using the local block matching method with the SSD\_N metric [1, 2] and the  $21 \times 21$  pixels window (Fig. 3c). Depth values were calculated only for pixels which belonged to the hand area (Fig. 3a i b). The hand area could be located with the thresholding method, because the hand was photographed against the dark background.

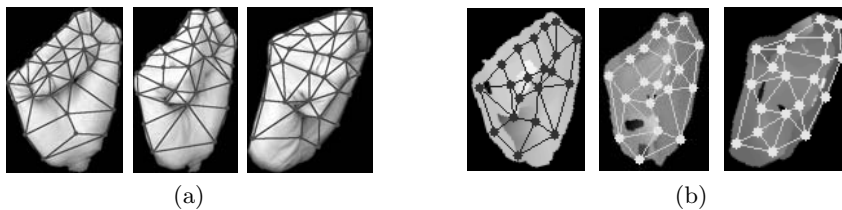


**Fig. 3.** Hand areas (b) detected for the pair of stereo images of Ch model posture (a) and the depth map obtained for this posture (c)

The first classifier was created on the basis of Gabor filters. The second one used the edge information and the third one depth maps. Graph models for the first and the second classifier were constructed on the basis of the model images of  $768 \times 576$  pixels resolution. They contained 35 nodes and 70 edges. The nodes were manually placed at anatomically significant points (Fig. 4a). Graph models created for the third classifier had 20 nodes, placed at the points representing particular parts of the hand, and 40 edges (Fig. 4b). During the first stage of the graph model matching in the recognition process it was possible to shift the graph's position by up to 30 pixels in  $x$  and  $y$  direction, at 5-pixel intervals, for the first two classifiers and up to 15 pixels (at a pixel interval) for the third classifier. It was also rotated by up to  $\pm 15^\circ$  for all classifiers, at  $5^\circ$  intervals for the first two classifiers and  $3^\circ$  intervals for the third classifier. At the second stage it was possible to shift the position of all nodes by up to 30 pixels for the first two classifiers and up to 6 pixels for the third one. The sizes of the surrounding area used for the jet's description were  $9 \times 9$ ,  $5 \times 5$  and  $2 \times 2$  pixels for the three successive classifiers, respectively. The costs coefficient was  $\lambda = 0.1$ .

The database test images classification results obtained for each classifier are presented in columns *Gabor*, *Edges* and *Depth*, respectively, of the Tab. 1. The classifiers based on Gabor filters and the edge information gave similar results.





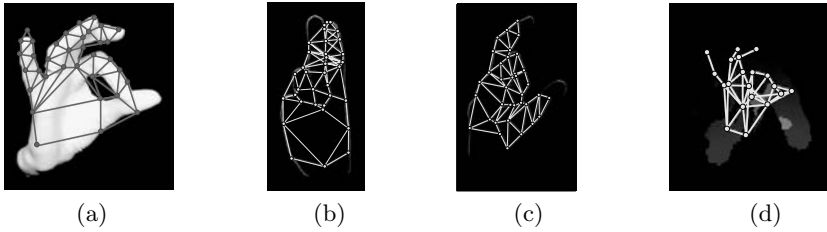
**Fig. 4.** A–posture graph models for the classifiers based on Gabor filters and the edge information (a) and depth information (b)

**Table 1.** Recognition results obtained for the PFA hand postures with the EGM method for different kinds of jets

Posture	Gabor		Edges		Depth		Hybrid	
	Correct	Errors	Correct	Errors	Correct	Errors	Correct	Errors
A	14	4D,1M,1P	14	1E,5P	14	4E,1F,1P	15	1D,2E
B	15	1D,1I,2K,1R	20		13	11,1K,4R,1Z	19	1R
C	9	4D,2L,1M,3N,1U	9	4D,6L,1U	12	1D,7L	12	1D,5L,1M,1N
Ch	12	1C,1D,1E,1M,1N,3U	15	2D,2E,1P	12	3A,1C,1D,1E,1L,1P	15	1C,2D,2E
D	19	1K	18	2K	15	1I,2K,1P,1S	19	1K
E	17	2A,1P	17	1A,2P	17	3P	19	1A
F	8	3D,2G,7S	16	1G,3S	16	3I,1Z	15	1G,1H,3S
G	17	2D,1P	15	5P	17	3A	19	1D
H	17	2D,1R	16	3D,1Y	17	1W,1Y,1Z	17	2D,1Y
I	16	2D,2Z	15	2D,3Z	20		19	1D
K	3	12D,5R	5	6D,8R,1Sz	9	1D,1H,1I,7R,1Z	7	5D,7R,1Sz
L	20		20		20		20	
M	19	1N	20		17	1E,1N,1P	20	
N	17	1D,2M	16	1L,3M	20		20	
O	9	1B,1F,1L,8S	12	1B,3F,4S	6	12F,2I	12	1B,3F,4S
P	16	4E	13	1D,6E	12	1A,7E	15	5E
R	7	13D	13	4D,2K,1Z	4	1I,3K,12Z	12	4D,1K,3Z
S	11	1F,2G,2L,4O	11	1F,1G,7O	7	2F,3I,8O	11	1F,8O
Sz	9	6D,1K,4R	12	1D,2K,5R	10	2D,2K,3R,1W,2Z	11	3D,2K,4R
U	15	4D,1N	17	3D	8	2M,1N,9S	18	2D
W	18	2R	18	2R	19	1H	18	2R
Y	17	3K	19	1K	13	5K,1R,1Z	20	
Z	20		18	2K	20		20	

The postures most often confused by all classifiers concerned the following groups of similar shapes: (a) F, G, O, S, (b) K, R, Sz, D, Z, (c) C, Ch, L. Within groups (a) and (b) it is the better matching of the nodes appearing at the no–differences–between–postures places that decides about the final result (Fig. 5a, b). The errors within group (c) result from large variations in hand shapes connected with the postures from these groups (Fig. 5c). Graph model requires great distortion tolerance to adjust to a given posture variations. By reducing topological costs big distortions become possible, but there are more false targets for every node. The depth based classifier gave the worst classification ratio. Apart from postures from groups (a) – (c) it also wrongly classified posture U (Fig. 5d). The reasons are the same as for postures from group (c).

Next experiment concerned hybrid classifier, which used output from three described above. Three the best classes occurring at the output of these classifiers were assigned points. The points were calculated as a product of two values. The first value corresponded to the position of the class at the output of the proper classifier. It was equal to 3, 2 or 1 if the class under consideration appeared at the first, second or third position, respectively. The second value was equal to



**Fig. 5.** An example of wrong classification: a) S classified as O – Gabor filters, b) and c) K as D and C as L – edge information, d) U as S – depth maps

normalized value of the recognition rate related to the class in given classifier. The class appeared at the output of the hybrid classifier was determined by the class which obtained the most points. An experiment on this classifier was performed using four runs of cross validation. For this purpose, test images from the database were divided into four subsets. The results are presented in the column *Hybrid* of the Tab. 1. The highest recognition ratio (80.9%) was obtained using this classifier. Relevant values for 3 former classifiers were 69.1%, 75.9% and 70.7%. Using hybrid classifier 5 postures were recognized better, and 10 postures not worse than by the best constituent classifier.

## 4 Conclusions

In the present paper the results of applying the EGM method to the 23 PFA postures classification were presented. Three different kinds of jets were used in three examined classifiers. The next classifier was based on the previous three ones. Experiments were carried out on the image database consisting of 460 test images. They proved that the EGM method can be used to assign the PFA postures to the specified classes. Additional analysis is required to distinguish similar postures, such as K, R and Sz. The sequence of images representing hand shape changes can be used for this purpose. An approximate shape of the examined posture can be obtained from the previous frames. It allows for the decrease of the number of graph models considered during classification. Additionally area in which the graph model is matched could be decreased. In this way the computation time can be reduced. The original EGM method cannot be used in real-time applications because Gabor transformation takes too much time. Using the features proposed in the present paper significantly speeds up the calculations. Research in this direction is planned for the future work.

## References

1. Brown M Z (2003) Advances in Computational Stereo. IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 25, No. 8
2. Cyganek B (2002) Komputerowe przetwarzanie obrazw trjwymiarowych. Akademicka Oficyna Wydawnicza ELIT, Warszawa

3. Grobel K, Hienz H (1996) Video-based recognition of fingerspelling in real time. Workshop Bildverarbeitung für die Medizin, Aachen
4. Hendzel J K (1995) Sownik Polskiego Jzyka Miganego. Wyd. OFFER, Olsztyn
5. Jiangqin W, Wen G (2001) The recognition of finger-spelling for Chinese sign language. Gesture Workshop, London, 96–100.
6. Lades M, Vorbruggen J C, Buhmann J, Lange J, von der Malsburg C, Wrtz R P, W. Konen W (1993) Distortion invariant object recognition in the dynamic link architecture. IEEE Transaction on Computers, Vol. 42, 300–311
7. Marnik J (2002) Rozpoznawanie znakw Polskiego Alfabetu Palcowego z wykorzystaniem morfologii matematycznej i sieci neuronowych. Rozprawa doktorska, Akademia Grniczo – Hutnicza, Krakw
8. Marnik J, Wysocki M (2004) Hand Posture Recognition Using Mathematical Morphology. Archiwum Informatyki Teoretycznej i Stosowanej, 16, 4, 279–293
9. Triesch J, von der Malsburg C (1996) Robust classification of hand postures against complex background. Proc. of the 2nd Intl. Conf. On Automatic Face and Gesture Recognition, Killington, Vermont, 170–175
10. Triesch J, von der Malsburg C (2002) A system for person-independent hand posture recognition against complex backgrounds. IEEE Trans. Pattern Anal. Machine Intelligence, 23(12): 1449–1453
11. Wiskott L, Fellous J-M, Krger N, von der Malsburg C (1995) Face recognition and gender determination. Int. Workshop on Automatic Face- and Gesture-Recognition, Zrich, 26–28

---

# Character Recognition Based on Fourier Transform and CDWT in Postal Applications

Mirosław Maszewski and Mirosław Miciak

University of Technology and Life Sciences in Bydgoszcz  
{mmasz,miciak}@utp.edu.pl

**Summary.** The article contains basic processing of the character image and calculation of characteristic features, on basis of which it will be recognized. The main objective of this article is to use Fourier transform and Complex Discrete Wavelet Transform to obtain a set of features which are invariant under translation, rotation, and scaling.

## 1 Introduction

The today's systems of automatic sorting of the post mails use the OCR mechanisms. In the present recognizing of addresses is insufficient. Therefore the OCR module is improving, particularly in the field of recognizing of handwritten characters. Although, these satisfactory results were received for printed writing, the handwriting is still difficult to recognize. The process of recognizing the writing, can be divided into: pre-processing and character recognition.

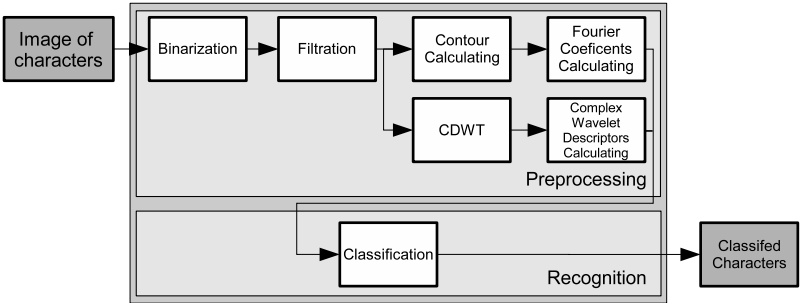


Fig. 1. Block diagram for the proposed system of character recognition

## 2 Binarization and Filtration

The colorful image must be converted to the image with levels of gray scale. The filtration of digital images is obtained by convolution operation. The new value of point of image is counted on the basis of neighboring points value. Every value is classified and it has influence on new value of point of the image after

filtration [1]. The applied filter is median filter, with mask  $3 \times 3$ . Then, the image is thresholding Otsu method [2], where a value of threshold level is chosen by minimization of variation in the class.

The outer boundary of digits is not always connected, therefore, a m-connectivity connection technique is used to connect boundary points of the digit [2] or use morphological operator: dilatation. This operation effectively repairs broken topological links resulting from poor image resolution or other image degradation. Dilation adds additional similar pixels around those which are deemed interesting [3].

### 3 Contour and Fourier Descriptors Calculating

The contour coordinate sequence  $x(m)$ ,  $y(m)$  can be calculated for each character from shape contour calculating stage. The character boundary tracking is a feature on basis of which the characters compare will be done.

$$r(t) = \left( [x(t) - x_c]^2 + [y(t) - y_c]^2 \right)^{1/2} \tag{1}$$

$$x_x = \sum_{t=0}^{N-1} x(t), \quad y_x = \sum_{t=0}^{N-1} y(t) \tag{2}$$

The Fourier series expansion of the co-ordinate sequences  $x(m)$  and  $y(m)$  are given by:

$$x(m) = \sum_{n=0}^{N-1} a(n) \exp(j2\pi nm/N) \quad y(m) = \sum_{n=0}^{N-1} b(n) \exp(j2\pi nm/N) \tag{3}$$

$$a(n) = \frac{1}{N} \sum_{m=0}^{N-1} x(m) \exp(-j2\pi nm/N) \quad b(n) = \frac{1}{N} \sum_{m=0}^{N-1} y(m) \exp(-j2\pi nm/N) \tag{4}$$

where:  $N$  is amount of point of contour  $a(n)$  and  $b(n)$  are the complex Fourier coefficients.

We can also obtain complex sequence  $c(n) = a(n) + jb(n)$  [4]. The Fourier coefficients  $a(n)$   $b(n)$   $c(n)$  are not rotation and shift invariant. In order to receive the features for character recognition stage may use the following coefficients (without complex phase information) [5]:  $|a(n)|$ ,  $|b(n)|$  and  $|c(n)|$ . It has been found [6] that calculating the coefficients  $|a(n)|$ ,  $|b(n)|$  for  $n = 1, 2, 3$  and  $|c(n)|$  for  $n = 1, 2, 3, N - 3, N - 2, N - 1$  are sufficient to correctly classify the main contour of a character. These coefficients contain factors dependent upon size which affects the recognition accuracy. Therefore these coefficients should be normalized in the complex frequency domain.

The  $x$ -axis is treated as the real axis, the  $y$ -axis is treated as the imaginary axis of a sequence of complex numbers. The FD coefficients for sequence  $s(k)$  are given by [4]. A close shape of digit boundary  $s$  can be expressed as  $(x_0, y_0)$ ,

$(x_1, y_1), (x_2, y_2), \dots, (x_{N-1}, y_{N-1})$ . These coordinates can be expressed in the form  $x(k) = xk$  and  $y(k) = yk$ . The boundary can be expressed as a sequence of  $s(k) = x(k) + iy(k)$ ,  $k = 0, 1, 2, \dots, N - 1$  [4].

$$z(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k)e^{-j2\pi uk/N}, u = 0, 1, \dots, N - 1 \tag{5}$$

The invariance to size, rotation, shift and start position of contour calculating can be achieved by use:

$$c(u - 2) = \frac{|z(u)|}{|z(1)|}, u = 2, 3, \dots, N - 1 \tag{6}$$

As a result of this stage is a  $n$  - dimensional feature vector including coefficients  $c(u)$ , where  $u = 2, 3, \dots, n - 2$ . In our experiment we chose  $n = 12$  for comparing results with other approach.

### 4 CDWT – Complex Discrete Wavelet Transform

In generally, it could be said that this transformation is an extension of classical DWT where real-valued discrete filters are replaced by filters with complex coefficients. However it is important to notice that there is no one way of CDWT realization, but it can be said that under that term there are many different and independent ways of getting the basis of CDWT, and that there exist many independent ways of practical realizations of CDWT. Fundamentally, many different conceptions of CDWT that are known from literature can be divided into two main categories:

The first category consist of wavelets that are used in applications where precise or fast signal (image) analysis is needed in purpose to distinguish characteristic elements and objects that are placed in the analyzed signal.

*Gabor Wavelets* based on pyramidal decomposition of signal with use of scaled and dilated kernel of Gabor transform belongs to this category. Changes of parameters of Gabor window function allows to generate large family of 2-D filters with large numbers of directional orientations [12, 13, 15, 14, 16, 18, 17].

*Magarey Wavelets* are the second type of CDWT that belongs to this category, which use modified algorithm of DWT transformation, mainly for purpose of fast and precise directional analysis of information. The characteristic features of these wavelets are very short digital filters, usually with 4 coefficients, for which all computations can be done with integer arithmetic only [19, 21].

Unfortunately, for all kinds of CDWT that belong to this category, there do not exist easy and fast algorithms of inverse transformation, that allow to reconstruct the original signal based on set of coefficients. This fact makes those transformations suitable mainly for image analysis applications with necessity of keeping the processed images in original form.

The second category of CDWT consist of whole family transformation for which one of the criterion of creation was the possibility of perfect reconstruction of

decomposed signal from set of coefficients to original signal form or to near original (estimated) form with very small difference.

*Symmetric Orthogonal Complex Daubechies Wavelets* belong to this category, which came into being if in the phase of projection of digital transformation filters, there is no restriction to real value of the calculated coefficients. The complex valued coefficients are then directly applied in the analysis-synthesis algorithm of Discrete Wavelet Transformation [22, 23, 24, 20, 25, 26].

A different ways for generating of the coefficients of CDWT, represent algorithms of reversible CDWT based on structure that is usually called *Dual Tree*. The main aim of authors of this conceptions was achieved the form of complex transformation with possibility of perfect reconstruction which is realized by digital filtration of signal only with use of filters with real valued coefficients. The two trees of wavelet decomposition represent together two sets of coefficients. When this coefficients are grouping by pairs then we poses a form of complex numbers [27, 28, 29, 30, 31].

The third conception is based on transformation of the real-valued signal to *analytic form* with use of the *Hilbert Transform*. After that process, signal obtained in that way is decomposed with use of any, well known sets of real-valued DWT filters that allow for perfect reconstruction. The real-valued signal, after that kind of complex projection, has complex-valued form, so at each level of real-valued DWT decomposition, a complex-valued representation of the decomposed signal is obtained [32, 33, 34].

### 5 Complex Wavelet Descriptors Calculating

For creating the feature vector of the recognized object in proposed character recognition system, set of values computed based on distribution of energy of decomposed signals for different subbands of complex discrete wavelet transform it was used. This values can be named as *"Complex Wavelet Descriptors"*.

For each of subimages, received from CDWT, the Normalized Wavelet Energy Signatures – *ZFS* are calculated, which are described by equation (7).

The *"pattern"* vectors were calculated based on detail subimages of *4-levels* CDWT of images from training system sequence. For each of particular detailed subimage, vector of features consist of three values of energetic wavelet signatures was calculated.

$$ZFS = \frac{1}{NM} \begin{bmatrix} SR_{j,k}^d = \sum_{n_x=0}^{N-1} \sum_{n_y=0}^{M-1} C_{j,k}^d(n_x, n_y), \\ OD_{j,k}^d = \sqrt{\sum_{n_x=0}^{N-1} \sum_{n_y=0}^{M-1} (C_{j,k}^d(n_x, n_y) - SR_{j,k}^d)^2}, \\ E_{j,k}^d = \sum_{n_x=0}^{N-1} \sum_{n_y=0}^{M-1} (C_{j,k}^d(n_x, n_y))^2 \end{bmatrix}, \quad (7)$$

As a result of this computation a vector of 12 or 24 values is received<sup>1</sup>. For each of base image, the received feature vector was stored in data base as "character complex wavelet descriptor".

This method, in its original form, is based on global energetic parameters calculation for subimages. It is proposed to extend this idea for some modification that allows to analyze local proprieties of the particular subimages. Each of subimages is divided into tiles of constant shape.

## 6 Classification

The classification in the OCR module compared features from the pattern to model features sets obtained during the learning process. Based on the feature vector, the classification attempts to identify the character based on the calculation of Euclidean distance between the features of the character and of the character models [7]. The distance function is given by:

$$D(C_i, C_r) = \sum_{j=1}^N [R(j) - A(j)]^2 \quad (8)$$

where  $C_i$  - is the predefined character,  $C_r$  - is the character to be recognized,  $R$  - is the feature vector of the character to be recognized,  $A$  - is the feature vector of the predefined character,  $N$  - is the number of features.

## 7 Experimental Results

The effectiveness and reliability assessment of proposed system, was calculated based on statistical analysis of achieved classification result. As an experiment's material for all tests an image database of handwritten digits and postal addresses was used. The selection of research materials, was conditioned by possibilities of practical implementation of proposed system in postal application for analysis, identification and distribution of messages, post letters and parcels.

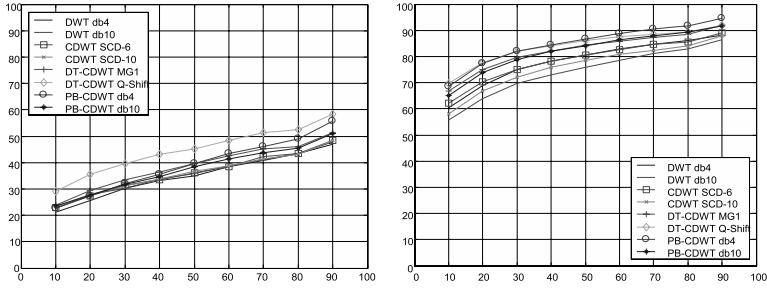
**Experiment's scope** – evaluation of the efficiency level in order of successful classification of analyzed image of character into one of classes which represent digit from '0' to '9', letters from 'A' to 'Z' and special characters like '-', ',', '.' etc.

**Experiment's scenario** – random division of samples from image database for training set and testing set. Participation of each part was changed from 10% to 90% with 10% steps. Each random variant was generated 100 times. For each of the proposed methods full process of transformation and generation the binary detail masks and sets of feature vectors was done, based on 4 – levels complex discrete wavelet decomposition with realization:

- CDWT – based on Symmetric Daubechies wavelets type SCD-6 i SCD-10,

<sup>1</sup> The number of elements in feature vector depends on the kind of CDWT.





**Fig. 2.** efficiency factor of correct classifications based on complex wavelet descriptors in: global analysis variant and local analysis variant with "16 tiles"

- DT-CDWT – based on MG1 and Q-Shift ideas,
- PB-CDWT – based on projected DB-4 and DB-10 wavelets.

Additionally, the received results were compared with those which are received by classical real-valued DWT transformation based on DB-4 and DB-10 wavelets. The achieved results of correct classification efficiency factor for this experiment is presented in figure 2.

## 8 Summary

Comparing results for handwritten character with other researches is a difficult task because are differences in experimental methodology, experimental settings and handwriting database. Liu and Sako [8] presented a handwritten character recognition system with modified quadratic discriminant function, they recorded recognition rate of above 98%. Kaufman and Bunke [9] employed Hidden Markov Models for digits recognition. They obtained a recognition rate of 87%. Aissaoui [10] using Normalized Fourier Descriptors for character recognition, obtained a recognition rate above 96%. Bellili using the MLP-SVM recognize achieves a recognition rate 98% for real mail zip code digits recognition task [11]. In our experiment we obtain recognition rate 79% for digits without broken or separated elements.

Based on received results for classification experiment with use of CDWT, we can see that the simplest energy signature method without tiles gives unsatisfactory results even below 50%. In case of image division on *tiles* for the first method the efficacy of recognition increases significantly (about 30%). When we divide image for more tiles efficacy of recognition increases. Of course, there are constraints in number of tiles. We examined that in our case more than 12 tiles do not give us significant improvement. The best results for this method (correct classification degree over 90%) were achieved when DT-CDWT in Q-Shift version and PB-CDWT – db4 was used.

**Table 1.** Examples of normalized Fourier coefficients for digits 3 and 8 from the database

Digit/ coefficients	1	2	3	4	5	6
3	0,24758	0,14477	0,029003	0,058297	0,016983	0,008532
8	0,341817	0,009269	0,044921	0,006339	0,017448	0,005609
Digit/ coefficients	7	8	9	10	11	12
3	0,015462	0,006949	0,01016	0,002727	0,005537	0,003224
8	0,004185	0,002814	0,00281	0,002039	0,001106	0,00198

## References

1. Ruminski J (2002) Metody reprezentacji, przetwarzania i analizy obrazw w medycynie. Politechnika Gdanska, Gdansk
2. Gonzales RC (2002) Digital Image Processing. Prentice Hall, New Jersey
3. Rashid T (1999) Clustering of Fuzzy image Features. Mathematics Department, Macquarie University, Sydney
4. Qinq C (2003) Optical Character Recognition for Model-based Recognition Applications. University of Ottawa, Canada
5. Zhu X (2000) A criterion based on Fourier transform for segmentation of connected digits. Tsinghua University, China
6. Rashid T (1999) Clustering of Fuzzy image Features. Mathematics Department, Macquarie University, Sydney
7. Günsel B (1998) Content-Based Video Abstraction. Int. Conf. on IP., Chicago
8. Liu C, Sako H (2002) Performance evaluation of pattern classifiers for handwritten character recognition. Int. Jour. on DAR., Springer, New York
9. Kaufmann G, Bunke H (2000) Automated Reading of Cheque Amounts. Pattern Analysis and Applications, Springer, New York
10. Aissaoui A (1999) Normalised Fourier Coefficients for Cursive Arabic Script recognition. Universite Mohamed, Morocco
11. Bellili A, Giloux M (2003) An MLP-SVM combination architecture for handwritten digit recognition. Int. Jour. on DAR., Springer, New York
12. J. Daugman (1980) Two-Dimensional Spectral Analysis of Cortical Receptive Fields Profiles. Vision Research, An Int. Jour. for FAV, ELSEVIER
13. J. Daugman (1988) Complete Discrete 2-D Gabor Transform by Neural Networks for Image Analysis and Compression. IEEE Tran. on ASSP., IEEE Press
14. B.S. Manjunath, W.Y. Ma (1996) Texture Features for Browsing and Retrieval of Image Data. IEEE Tran. on PAMI., IEEE Press
15. S. Marcelija, (1980) Mathematical Description of the Responses of Simple Cortical Cells. Jour. of the OSA, OSA Publishing
16. T. Andrysiak, M. Choras (2003) Hierarchical Object Recognition Using Gabor Wavelets. Proc. of CRS. Conf. – KORES'03, Mikow
17. T. Andrysiak, M. Choras (2005) LPT and DST for Face Feature Extraction and Representation. Proc. of IAPR Conf. on PRIP – PRIP'05, Misk
18. T. Andrysiak, M. Choras (2005) Face Feature Extraction and Representation. Proc. of ACS-CISIM, vol. 1, Biaystok
19. J.F.A. Magarey (1997) Motion Estimation using Complex Wavelets. PhD Thesis, Department Of Engenering, Cambridge University

20. J.M. Lina (1997) Image Processing with Complex Daubechies Wavelets. Jour. of MIV, Springer-Verlag
21. J.F.A. Magarey, N. Kingsbury (1998) Motion Estimation using Complex-valued Wavelet Transform. IEEE Tran. on SP., IEEE Press
22. W. Lawton (1993) Applications of Complex Valued Wavelet Transforms to Subband Decomposition IEEE Tran. on SP., IEEE Press
23. J.M. Lina, M. Mayrand (1995) Complex Daubechies Wavelets. Jour. of ACHA., ELSEVIER
24. J.M. Lina (1996) Complex Daubechies Wavelets: filters design and applications. Report No. 2449, CRM, <http://citeseer.ist.psu.edu/lina97complex.html>
25. L. Gagnon, J.M. Lina (1994) Symmetric Daubechies' wavelets and numerical solution of NLS equations. Jour. of Physics A, IOP Publishing Ltd.
26. X.P. Zhang, M. Desai, Y.N. Peng (1999) Orthogonal Complex Filter Banks and Wavelets: Some Properties and Design IEEE Tran. on SP., IEEE Press, <http://citeseer.ist.psu.edu/zhang99orthogonal.html>
27. N. Kingsbury (1999) Shift Invariant Properties of the Dual-Tree Complex Wavelet Transform. Proc. of IEEE Conf. ASSP., Phoenix
28. N. Kingsbury (2001) Complex Wavelets for Shift Invariant Analysis and Filtering of Signals. Jour. of ACHA, ELSEVIER
29. N. Kingsbury (2003) Design of Q-shift Complex Wavelets for Image Processing using Frequency Domain Energy Minimisation. Proc. of IEEE Conf. on IP., Barcelona
30. I.W. Selesnick (2001) Hilbert Transform Pairs of Wavelet Bases. IEEE SP. Letters, IEEE Press
31. I.W. Selesnick (2001) The Double Density Dual-Tree Discrete Wavelet Transform IEEE Tran. on SP., IEEE Press
32. F. Fernandes (2002) Directional, Shift-insensitive, Complex Wavelet Transform with Controllable Redundancy PhD Thesis, Rice University, Houston
33. F. Fernandes, I.W. Selesnick, R. Spaendonck, C. Burrus (2003) Complex Wavelets Transform with Allpass Filters. Proc. of SP., Zagreb, Croatia
34. F. Fernandes, R. Spaendonck, C. Burrus (2003) A New Framework for Complex Wavelet Transforms. IEEE Tran. on SP., IEEE Press

---

# Preprocessing for Real-Time Handwritten Character Recognition

Bartosz Paszkowski, Wojciech Bieniecki, and Szymon Grabowski

Katedra Informatyki Stosowanej, Politechnika Łódzka  
wbieniec@kis.p.lodz.pl

**Summary.** We present a real-time on-line handwritten character recognition system, based on an ensemble of neural networks. In this work we focus on the developed preprocessing algorithms which help achieve high accuracy rate without a visible delay in recognition process.

## 1 Introduction

An important trend in technology today is mobile computing. Among the main challenges, however, with making portable devices truly useful for a wide range of applications are the user interface issues: how to enter data in a convenient way and how to present information on a small display. Many PDA devices (palmtop, Tablet-PC, smartphone, etc.) got rid of a traditional keyboard and let users manually write with a stylus on a touch screen. Efficiency and effectiveness of such a solution are vitally dependent on the robustness of the underlying recognition system. Unfortunately, the accuracy of character recognition (CR) systems (e.g., Calligrapher for Pocket PC [2]) implemented in those devices is still imperfect.

We conjecture that one of the most important components of a handwritten character recognition system is its preprocessing stage. The application we present in the current work is based on a very simple classifier, a single layer perceptron (SLP), still, the overall recognition accuracy is quite satisfactory, thanks to the preprocessing techniques we use.

Moreover, our system works in real time, trying to guess the entered character after each pen stroke. Once a character is drawn, it can be corrected or altered (e.g. digit 9 into 8) by the writer, and then immediately the system “notifies” it has to process the same character again, and reacts appropriately. We find this feature user friendly.

## 2 Related Work

The origins of automatic recognition of handwritten characters date back to 1940s, but rapid development had been observed in 1980s and early 1990s, mostly due to the progress in powerful computing hardware and image acquisition devices.

An excellent survey of handwritten CR techniques can be found in [1]. The authors identify five major phases of the CR problem: preprocessing, segmentation, representation, training and recognition, and postprocessing. Our work focuses on the preprocessing, but all the other stages do exist as well in our application, in one form or another.

Tappert [5] defines five difficulty levels for handwritten CR: boxed discrete characters, spaced discrete characters, run-on (touching or overlapping) discretely written characters, pure cursive handwriting, and mixed cursive and discrete characters. The first, easiest, problem variant is the only one which does not require character segmentation. The recognition task we deal with belongs to the third category, discretely written, but possibly overlapping, characters.

### 3 Requirements for a Real-Time Handwritten Character Recognition Application

We set the following requirements for our application.

- High accuracy. According to [6], people find handwriting recognition software useful only if it recognizes over 97% to 98% of the characters.
- The recognition should be robust to character translation, their rotation and scale, character slant, random distortions (e.g., shaky writing), right- or left-handedness of the writer, and so on.
- Real-time recognition even on low computing power machines.
- Characters are written with a pointing device (e.g. pen, mouse),
- Characters are recognized individually, the recognition output is shown on screen immediately, after each entered character.
- The set of output classes is fixed. In the current version of the system there is no “reject” or “uncertain” answer.
- Characters are glued into words which are subsequently passed to a dictionary of valid words; unrecognized words are marked (e.g., underlined).
- The application should work in the following modes:
  - $[a - zA - Z0 - 9]$  characters recognized; dictionary on.
  - $[a - zA - Z0 - 9]$  characters recognized; dictionary off.
  - Digits only.
  - Capital letters only (dictionary on or off).
  - Lowercase letters only (dictionary on or off).

Using a restricted alphabet should improve classification accuracy and speed. Also the preprocessing can be made faster if some preprocessing steps are specific only for a class of characters.

The main problems to solve to meet the listed requirements are:

- finding an appropriate character’s bounding box resolution, to protect from low accuracy and low processing speed at the same time,
- training set generation (robust for many individual handwriting styles),

- segmentation (grouping line segments, perhaps disjoint, into characters, and characters into words),
- devising low-complexity algorithms and implementing them with speed optimizations (crucial for the preprocessing algorithms).

## 4 Handwriting Recognition Engine: Preprocessor

Roughly speaking, the task of a preprocessor in an OCR system, is to segment characters, correct their defects, remove excessive information, and convert into a form that can be feasibly processed by the classifier.

In our system, the preprocessor is built of several modules, run one by one: excessive line segment reduction, rotation removal, scaling, translation, bitmap generation, noise removal and line thinning, and generating an input vector for the classifier.

Some of those operations are quite trivial (e.g., scaling the character to fit a predefined bounding box, translating the character image to be anchored at the origin), and we do not describe them in detail.

### 4.1 Excessive Line Segment Reduction Module

The module input is the order set of line segments of a written character. What the module returns is a simplified description of the character. More concretely, its output is the character composed of a minimal set of line segments which approximate the original set of segments with user-defined accuracy. Our algorithm is a modification of Jenks' algorithm [4].

The algorithm's input is a character represented as a set of  $n$  line segments,  $l_i$  for  $i \in \{0 \dots n - 1\}$ , ordered adjacently (one segment's end point is the start point of the next segment), and a reduction parameter  $r$  from  $[0; 1]$ , a higher value means a stronger reduction. Assume also that  $l_n$  is another way to denote  $l_0$ . Let  $w$  and  $h$  denote the character bitmap width and height, respectively. Let  $max = maximum(w, h)$ . For convenience, we use a variable for reduction distance,  $rd$ . Let  $rd = max * r$ .

Now we run the following procedure.

```
repeat
  for i: 0...n-1 do
    create templine from start point of l_i to end point of l_{i+1}
    if dist(start point of l_i, end point of l_{i+1}) <= rd then
      l_i = templine
      remove line_{i+1}
    end if
  end for
until no line removed
```

We set  $r = 0.01$  for the tests. The number of iterations over line segments is 4 on average, in practice. The amount of eliminated segments varies from a few to about 80% in some cases (slow writing typically generates more segments).

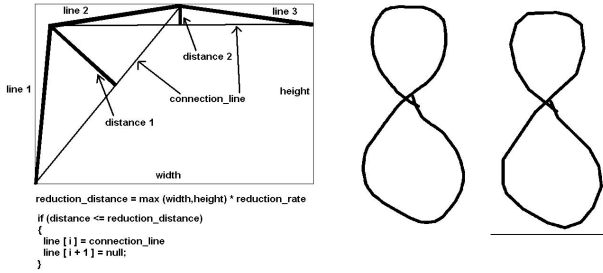


Fig. 1. Jenks algorithm

### 4.2 Rotation Removal

The input character image may be rotated. We use a very simple algorithm to remove this defect. It starts with scaling a character to have its height (at least slightly) larger than its width. Then, it rotates the bounding box of a character many times, in the angle range  $\{-89 \dots 89\}$ , with 2 degree angle difference, and outputs the image for whose the bounding box width was lowest.



Fig. 2. The result of using scaling (on the left), and rotation removal (on the right) algorithms

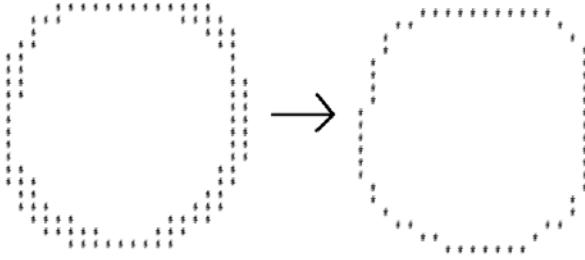
### 4.3 Bitmap Generation

So far, a character has been represented as a set of line segments (of some thickness). Further steps require having a bitmap image. Such a lossless conversion would be trivial, but the current module does a little more: it also joints disconnected line segments if they are close enough.

### 4.4 Noise Removal and Line Thinning

The input image may contain some isolated pixels, they are eliminated. Then we reduce the lines to obtain the character outline, using the thinning technique. Thinning is a morphological operation, somewhat related to erosion or opening. This module reduces all lines to have single pixel thickness. The resulting outline of an exemplary character is presented in Fig. 3.

It is remarkable that this algorithm creates the shape contour in a single iteration. It was achieved thanks to using appropriate masks applied in a proper order. Working in a single pass is, of course, a highly desirable feature for a real-time system.



**Fig. 3.** The result of using contouring algorithm

#### 4.5 Generating an Input Vector for the Classifier

The obtained bitmap is read column-wise and a bit vector  $B$  is achieved (where 0's correspond to background pixels). A row-wise scan could in fact be an alternative, as we have not notice any advantage of one variant over the other.

Then we create a vector  $V$  storing the positions of set bits in  $B$ , in their increasing order. This is a compact representation since the amount of pen pixels compared to the background pixels is typically (very) small. The  $V$  vector becomes an input pattern for the classifier.

### 5 Handwriting Recognition Engine: Classifier

The heart of any CR system is the classifier. We chose to use an ensemble of very simple classifiers, single layer perceptrons (SLP) [3], over majority voting. The ensemble diversity is obtained via random initial weights from the range  $< -0.5 \dots +0.5 >$ . Using voting over many individual SLP decisions avoid the “curse” of linear classifiers (and SLP does, of course, belong to this category), which cannot cope with complex decision boundaries. On the other hand, the classification speed of each SLP component is high enough not to prevent us from using tens (or even hundreds) components in parallel. In fact, memory scarcity in handheld devices may be more restrictive.

The number of inputs of each SLP is equal to the number of character pixels (same for each character, which is guaranteed by the scaling preprocessing operation). The number of output values, i.e., the number of class labels, is known beforehand for a given problem instance (for example, if the user selects the mode of recognizing digits only, the number of classes is 10).

We cannot provide details on training our SLP ensemble due to shortage of space.

## 6 Experimental Results

The experiment was set up in the following way:

**Creation of pattern sets:** one for digits and one for letters. Each digit was painted 70 times and a letter 45 times. Then the images have been artificially



distorted (rotation in a range  $< -85 \dots +85 >$  degrees and shear in a range of  $< -0.3 \dots +0.3 >$  in both vertical and horizontal directions).

**Each set was split** into a training set (2843 digit patterns and 2008 letters) and a test set (6633 digits and 4688 letters).

**Extension of the test sets** by adding extra distorted images ( $< -30 \dots +30 >$  degree rotations and  $< -0.03 \dots +0.03 >$  shears) to original patterns. After these operations we got a test set of 21856 digits and a test set of 15431 letters.

**Preprocessing of training sets.** This made all the patterns from training sets as simple as possible, so that the SLP could learn them without a big risk of getting stuck in local minima. That means that by using our preprocessor we expected to make all the training sets linearly separable or almost linearly separable. As a verification, two SLPs have been designed: one for digits and another for letters.

**Learning of the networks.** The process lasted only 9 seconds for both networks and showed that none of networks got stuck in local minimum. The first network learned all patterns (100%) of the training set and the second network learned 2003 patterns (99.75%).

**Testing of the networks.** The first network misclassified only 150 from 21856 patterns of digits and the second network misclassified only 335 from 15096 patterns of letters. Those results showed that the preprocessor fulfilled our expectations and managed to make almost linear-separable sets of patterns. The generalization quality measured on the test sets for first network was 99.31% and for the second network 97.83% (Table 1).

**Table 1.** The results of learning SLP (Time of learning was about 10s and the bitmap was 10x18.)

	Trainset size	Learned trainset samples	Testset size	Testset accuracy
Network (digits)	1 2843	2843 (100.00%)	21856	21706 (99.31%)
Network (letters)	2 2008	2003 (99.75%)	15431	15096 (97.83%)

## Working in Real-Time

The system has been implemented with Java language in JDK 1.6.0 version. The application was tested on a PC with Pentium 4 3GHz. It needs about 30MB of free operating memory to run (most of it for JVM). In Table 2. below, the average time of single character recognition has been shown:

These results show, that the system can work in a real-time mode. We didnt noticed any delays while testing the application in practice. The recognition results seem to be immediate for user.

**Table 2.** Average time of single-character recognition for different processors

	P4 3 GHz	Duron 800	P3 450 MHz
Avg time of single- char recognition	2ms	8ms	15ms

## 7 Conclusions

The implemented system has some **limitations**. It can recognize only non-connected handwritten characters and doesn't recognize scanned handwriting. The **most important component** of our implementation are: line simplification with a modified Jenks' algorithm, character auto-rotation algorithm, robust algorithm for merging close but disjoint line segments within individual writing strokes, a fast and reliable one-pass contouring, and a fast and useful routine helping to generate test sets, multiplying a human-generated image into many variants via affine transforms.

**In the future** we intend to introduce the following modifications:

- Handling more geometry distortion types
- Java 2 ME implementation for mobile, low speed devices
- instant presenting an uncertainty estimation
- online learning.

Although the recognition results are reasonably good, introducing other features (i.e. shape descriptors, correlation factors, image frequency analysis, line descriptors) may increase the reliability.

## References

1. Arica N., Yarman-Vural F. T. (2001): An Overview of Character Recognition Focused on Off-line Handwriting. Systems, Man and Cybernetics, Part C: Applications and Reviews, Vol. 31, No. 2, pp. 216-232.
2. PhatWare Corporation, CalliGrapher 8.5 for Pocket PCs, <http://www.phatware.com/calligrapher/>.
3. Duch W., Korbicz J., Rutkowski L., Tadeusiewicz R. (2000): Sieci neuronowe. PAN Akademicka Oficyna Wydawnicza EXIT, Warszawa.
4. Jenks G. F. (1989): Geographic logic in line generalization. Cartographica, 26(1): 27-42.
5. Tappert C. C. (1984): Adaptive on-line handwriting recognition. in Proc. 7th Int. Conf. Pattern Recognition, Montreal, Canada, pp. 1004-1007.
6. CharacTell Advanced Character Recognition Technology: <http://www.charactell.com/ACRWhitePaper.pdf> and <http://www.charactell.com/TechnologyFoundation.html>

---

# Handwritten Word Recognition with Combined Classifier Based on Tri-grams

Jerzy Sas<sup>1</sup> and Andrzej Zolnierrek<sup>2</sup>

<sup>1</sup> Wroclaw University of Technology, Institute of Applied Informatics, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland  
jerzy.sas@pwr.wroc.pl

<sup>2</sup> Wroclaw University of Technology, Faculty of Electronics, Chair of Systems and Computer Networks, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland  
andrzej.zolnierrek@pwr.wroc.pl

**Summary.** In the paper a new method of handwritten word recognition is described. In this method different probabilistic character language models (PCLMs) are used in order to improve the word recognition accuracy. These models consist of conditional probability distributions of characters given that preceding, succeeding or surrounding characters are known. The application of character succession and precedence in word recognition was explored in many earlier works. The novelty of the method proposed here consists in utilizing also the conditional probabilities based on character appearing of both sides of the letter being recognized and in combining such classifier with other ones based on character precedence and succession. This new classifier that uses two sided character neighborhood is realized as an iterative procedure which starts with character support factors evaluated independently for each character in the word by a soft character classifier. Then in each step of this iterative procedure the support factors are calculated as Bayes-like formula using needed probabilities. Next, the support factors obtained at the character level are used to calculate the support factors for complete words. Finally, the soft word recognition results obtained for classifiers based on three different PCLMs are combined. Experiments described in the paper show the superiority of the combined method over all its simpler components.

## 1 Introduction

Pattern recognition algorithms using context were successfully applied to the problem of hand printed letters classification ([1], [3], [5]). Moreover, for every language taking into account the characteristic dependences in the sequence of characters, the algorithms of words recognition were proposed. In such algorithms, the probabilistic character language model (PCLM) assuming Markov dependences usually was used. In such case either we can explore Bayes algorithm for Markov chains [9] or we can use hidden Markov (HMM) model [6] in pattern recognition algorithms construction. Let us notice, that under certain assumptions both methods are equivalent. In this paper we use the second of above mentioned probabilistic approach to the problem of word recognition, although some ideas of the first approach also will be taken into account. The

details of second-order HMM construction for words recognition supported by bi-gram and tri-gram character models are presented in [8]. In this paper except the typical algorithm using assumption of second-order Markov dependence in the sequence of letters in recognized word two different algorithms (using another PCLMs) are proposed. The first one is the simple modification in which we also assume the second-order Markov dependence but backward, however in the second algorithm the quite new model of dependence is proposed. In this model we assume that probability of every character in the word depends on preceding and following character ( we called it inward prediction). Next, these three classifiers will be combined in order to achieve the algorithm of better quality, i.e. in which the average error rate is less than obtained for every classifier separately. The organization of the paper is as follows: after introduction the problem statement is presented in second section. Next, the description and the probabilistic properties of every model of word recognition are presented. Consequently, in the section 4, new algorithm, based of inward prediction is described in details. Then, in section 5, the combined threefold algorithm is proposed. In section 6 the results of empirical investigation are presented and in the end we conclude this paper.

## 2 Problem Statement

The problem of word recognition can be treated as two-level classification problem. On the lower (character) level separate letters are being classified. On the upper (word) level the results of letter classifications are used to recognize the whole word. The word is hand printed, i.e it is written in block capital letters, enclosed in the separated regions (character fields) located in fixed positions on the printed form. Hence, it is easy to extract isolated images of subsequent characters and the problem of text segmentation is almost trivial and not discussed here. We also assume that the text image analyzer can perfectly distinguish between empty and nonempty character fields, so the length  $M$  of the word is fixed before start of recognition procedure. Let us consider the problem of correctly segmented word recognition. The word consists of  $M$  characters, each of them belonging to finite alphabet  $\mathcal{C} = \{c_1, c_2, \dots, c_L\}$ . The word image is segmented into  $M$  parts  $(x_1, x_2, \dots, x_M)$  containing consecutive character images. On the character level, isolated characters  $c_i, i = 1..M$  are recognized independently by a soft classifier  $\Phi$  [2] which gets character image  $x$  as its input and evaluates the vector of support factors for all characters in the alphabet, i.e.:

$$\Phi(x) = [d_1(x), d_2(x), \dots, d_L(x)]^T. \quad (1)$$

Without loss of generality we can restrict  $d_i(x)$  within the interval  $[0, 1]$  and additionally  $\sum_i d_i(x) = 1$ . Crisp classification  $\bar{\Phi}(x)$  can be easily obtained from the support vector by selecting this character, for which the support factor has the greatest value.

On the word level we can distinguish three sources of information, which can be used by the word classifier, namely:

- results of soft character classification for particular characters constituting the word on the character level ( $\Phi(x_1), \Phi(x_2), \dots, \Phi(x_M)$ ) or alternately: results of crisp character classification ( $\bar{\Phi}(x_1), \bar{\Phi}(x_2), \dots, \bar{\Phi}(x_M)$ ),
- confusion matrix of character recognizer  $E_{L \times L}$  in which every element  $e_{i,j} = p(\bar{\Phi}(x_j) = c_i | c_j)$  is a conditional probability that decision of character recognizer is  $c_i$  while the actual character on the image  $x_j$  is  $c_j$ .
- adopted probabilistic character language model. In this paper we take into account three different PCLMs. Each PCLM consists of:
  - probability distribution of leading and trailing characters (i.e. character  $s$  beginning and ending words)  $P_I = (p_1^I, p_2^I, \dots, p_N^I)$  and  $P_E = (p_1^E, p_2^E, \dots, p_N^E)$ , and corresponding set of probabilities:
  - set  $A_{LxLxL}^S$  of character succession conditional probabilities, where  $a_{i,i-1,i-2}^S = p_S(c_i | c_{i-1}, c_{i-2})$  is the probability that the next character in a word is  $c_i$  provided that two preceding characters are  $c_{i-1}$  and  $c_{i-2}$ ,
  - set  $A_{LxLxL}^P$  of character precedence conditional probabilities, where  $a_{i,i-1,i-2}^P = p_P(c_{i-2} | c_{i-1}, c_i)$  is the probability that the preceding character in a word is  $c_{i-2}$  provided that two next characters are  $c_{i-1}$  and  $c_i$ ,
  - set  $A_{LxLxL}^Q$  of character neighborhood conditional probabilities, where  $a_{i,i-1,i-2}^Q = p_Q(c_{i-1} | c_{i-2}, c_i)$  is the probability that the current character in a word is  $c_{i-2}$  provided that preceding character is  $c_{i-2}$  while successive character is  $c_i$ ,

The probability distribution of leading and trailing characters and sets of succession, precedence, neighborhood conditional probabilities can be obtained by simple analysis of text corpus. It can be assumed that character succession, and precedence as well as neighborhood probability distributions are common property of the language in which recognized texts are written and they are domain invariant. Hence the texts in the corpus does not have to be from the area of interest. Large corpora of general texts in the language can be used to estimate sets:  $A^S$ ,  $A^P$ ,  $A^Q$  and probabilities:  $P_I$ ,  $P_E$  respectively.

### 3 Probabilistic Properties of Applied PCLMs

In this paper we will apply three above mentioned PCLMs to the word recognition problem. Hence, we propose three approaches: prediction of the next character using two preceding characters, prediction of the character using two next succeeding ones or predicting the character using its two neighbors. Two first approaches can be easily utilized in HMM model applied as a recognition tool. Unfortunately, the character neighborhood approach can not be used directly in HMM model, because this model fits only to the sequential left-to-right or right-to-left analysis. Before we start with application of considered PCLMs to the problem of word recognition we test their predictive power. The test will consist in application only *a priori* knowledge (i.e.  $P_I$  or  $P_E$  and  $A^S$ ,  $A^P$  or  $A^Q$ )

in character recognition for every PCLM. In such a case applying of Bayes algorithm leads to the following general formula:

$$c^* = \arg \max_{c \in C} p_R(c | c_a, c_b) \tag{2}$$

where  $c_a, c_b$  are two preceding, succeeding or neighbor characters and  $R$  denotes  $S, P$  or  $Q$  depending on chosen PCLM. Then the average error rate for every proposed above classifiers can be calculated using the following formulas:

$$E_S = \sum_{c_{i-2}=1}^L \sum_{c_{i-1}=1}^L (1 - \max_{c_i \in C} p_S(c_i | c_{i-1}, c_{i-2}))p(c_{i-1}, c_{i-2}) \tag{3}$$

for succession prediction,

$$E_P = \sum_{c_{i-1}=1}^L \sum_{c_i=1}^L (1 - \max_{c_i \in C} p_P(c_{i-2} | c_{i-1}, c_i))p(c_{i-1}, c_i) \tag{4}$$

for precedence prediction, and

$$E_Q = \sum_{c_{i-2}=1}^L \sum_{c_i=1}^L (1 - \max_{c_i \in C} p_Q(c_{i-1} | c_{i-2}, c_i))p(c_{i-2}, c_i) \tag{5}$$

for neighborhood prediction, respectively.

In equations (3), (4) and (5)  $p(c_{i-1}, c_{i-2}), p(c_{i-1}, c_i)$  and  $p(c_{i-2}, c_i)$  denote corresponding two dimensional joint probability distribution of characters in a word. These probabilities as well as all needed conditional probabilities can be easily estimated from the language corpus. In order to compare the probabilistic properties of applied PCLM, empirical investigations were made. Namely, Bayes algorithm (2) using succession conditional probabilities  $p_S(c_i | c_{i-1}, c_{i-2})$ , precedence conditional probabilities  $p_P(c_{i-2} | c_{i-1}, c_i)$  and neighborhood conditional probabilities  $p_Q(c_{i-1} | c_{i-2}, c_i)$  successively was applied to four different corpora of Polish texts. Three of them contain passages from Polish press writing while the fourth one is the set of short notes from authentic medical patient's records. The results of error rate in percent are presented in the table below.

The effectiveness of succession and precedence prediction is approximately on the same level for all kind of texts. The effectiveness of neighborhood prediction is higher about 20 % in relation to succession and precedence method. Another observation is that the effectiveness of prediction based on PCLMs for medical texts is higher than for press excerpts. Here also the average error rate is about 20 % lower ( avg. 53 % for literature texts vs. 45.7 % for medical texts). The observation following from these experiments leads to the presumption that character neighborhood conditional probabilities can be effectively used in order to improve the word recognition accuracy.

**Table 1.** The error rate of different classifiers

Corpus	Trigram occurrence count	Succession	Precedence	Neighborhood
press I	1140,693	62.3%	62.0%	52.7%
press II	600,045	62.5%	63.2%	52.9%
press III	413,383	64.2%	64.7%	54.3%
medical	8502,208	56.6%	56.5%	45.7%

## 4 Word Recognition Algorithm Based on Inward Prediction

The results of experiments described in the previous section indicate that the inward PCLM can be useful in construction of pattern recognition algorithm. As it was mentioned above, the recognition based on second-order HMM described in [8] can not be used in this case.

For this reason, in the case on inward prediction quite different heuristic recognition procedure is used. Similarly as in the case of HMM based word recognizers the soft recognition paradigm is also applied here [2]. It means that the application of the soft recognizer results in providing the support factors for all possible classes (in our case classes are words that can be recognized). Support factors correspond to the likelihood that the currently recognized image represents given word. Application of the soft recognition paradigm makes it also easy to combine the results of soft classification independently obtained for forward, backward and inward word classifiers.

The procedure proposed for inward recognition uses soft recognition results achieved by applying soft character classifier to all character images in the word. Let  $d_{i,j}^{(0)}$  denotes the support factor for the character  $c_i$  at  $j$ -th position in the word. Then, we calculate iteratively the support factors  $d_{i,j}^{(k)}$  for iteration index  $k = 1, 2, \dots$  according to the following Bayes-like formula:

$$d_{i,j}^{(k)} = d_{i,j}^{(0)} \sum_{i'=1}^L \sum_{i''=1}^L d_{i',j-1}^{(k-1)} d_{i'',j+1}^{(k-1)} p_q(c_i | c_{i'} c_{i''}) \quad (6)$$

In the proposed formula (6) the support factors  $d_{i',j-1}^{(k)}$  and  $d_{i'',j+1}^{(k)}$  can be treated as estimates of probabilities for characters  $c_{i'}$  and  $c_{i''}$  at the positions  $j-1$  and  $j+1$  correspondingly. Then, the probability of the character  $c_i$  at the position  $j$  is re-estimated using: estimates of character probabilities at adjacent positions and the conditional probabilities  $p_q(c_i | c_{i'} c_{i''})$  contained in inward PCLM as well as the Bayes formula for total probability:  $\sum_{i'=1}^L \sum_{i''=1}^L d_{i',j-1}^{(k-1)} d_{i'',j+1}^{(k-1)} p_q(c_i | c_{i'} c_{i''})$ . Additionally, as to prevent the convergence of likelihood calculated in subsequent steps to the values independent on initial estimates provided by the soft character classifier, in every step the total probability of a character  $c_{i,j}$  is multiplied by initial estimate  $d_{i,j}^{(0)}$ . This procedure terminates after  $k^*$  iterations.

The number of iterations has to be established experimentally, as to maximize the accuracy of the word classification algorithm based on calculated values of  $d_{i,j}^{(k)}$ . In our experiments  $k^* = 8$  was applied.

The soft word recognition algorithm consists in finding the set of  $N$  most likely words and in evaluating support values for selected words. Word classifier produces the set of pairs:

$$R^I = \{(w_1, d_{w_1}^I), \dots, (w_N, d_{w_N}^I)\}, \tag{7}$$

where  $w_l, l = 1 \dots N$  is the word and  $d_{w_l}^I$  is its support value. For every considered word  $w_l = (c_1^{(l)}, c_2^{(l)}, \dots, c_M^{(l)})$ , its support factor can be calculated as:

$$d_{w_l}^I = \prod_{j=1}^M d_{c_j^{(l)}}^{k^*} \tag{8}$$

### 5 Combined Threefold Algorithm

The results of soft recognition provided by the algorithm based on inward prediction can be combined with the results of the soft recognition achieved for forward and backward prediction. The combined algorithm based only on forward and backward prediction using second-order HMMs, was proposed in [8]. The sets  $R^F$  and  $R^B$  are results of word recognition supported by forward and backward prediction and contain the most likely words with their support factors, similarly as defined in (7). The sets of words appearing in  $R^F, R^B$  and  $R^I$  can be different, but the most likely words usually appear in all three sets. Let  $d_w^F$  and  $d_w^B$  denote the support factors for the word  $w$  in  $R^F, R^B$  correspondingly.

The result of the combined threefold word recognition algorithm is the set  $R^{FBI}$  consisting of the most likely words with their support factors  $d_w^{FBI}$ . The set  $R^{FBI}$  is created in the following steps:

- let the set  $Z^{FBI}$  of words recognized by combined algorithm will be the intersection of word sets  $Z^F, Z^B$  and  $Z^I$  appearing in  $R^F, R^B$  and  $R^I$ ,
- calculate support factors  $d_w^{FBI}$  for all words in  $Z^{FBI}$  as the product  $d_w^F * d_w^B * d_w^I$ ,
- normalize factors obtained in this way so as to sum up to 1.0 over the whole set  $Z^{FBI}$ .

### 6 Experimental Results

The aim of experiments carried out was to investigate the accuracy of word recognition achieved by the classifier based on inward prediction and to evaluate the accuracy of the combined classifier. Because the work described here is a part of wider project related to medical handwritten document recognition, the text corpus consisting of excerpts from the authentic medical documentation was used. As the character classifier, MLP based recognizer was used based on



directional features. The classifier construction is described in [4] and [7]. The accuracy of the crisp character classifier obtained from the soft one is 93.4%.

The aim of the first experiment was to test how the word recognition accuracy depends on the number of iteration of the algorithm based on inward prediction. The word crisp recognition was performed using the support factors defined in (6) after  $k = 0...10$  iterations. For each iteration the word recognition accuracy was evaluated using the testing word set. The results are presented in Table 2:

**Table 2.** Word recognition accuracy as the function of iteration count of inward prediction algorithm

Iteration	0	1	2	3	4	5	6	7	8	9	10
Accuracy [%]	66.1	86.1	86.3	87.1	86.9	87.2	87.0	87.2	87.1	87.2	87.2

It can be observed that the recognition accuracy increases until fifth iteration and then it stabilizes on the highest level. For practical reasons it seems to be sufficient to execute 5...8 iterations.

In the next experiment the performance of the combined algorithm was compared with the uni-directional second-order HMM based word recognition algorithms. The accuracies of the compared algorithms are shown in Table 3.

**Table 3.** Word recognition accuracies achieved by simple and combined algorithms

Algorithm	forward	backward	inward	combined
Accuracy [%]	88.8	89.8	87.2	93.1

The combined classifier accuracy is significantly higher than the accuracy of each of its component classifiers. The error rate was reduced from 10.2% (in the case of the best component classifier) to 6.9%, what gives 32.5% of the relative error reduction.

## 7 Conclusions

Presented experimental results concerning proposed word recognition algorithm based on inward prediction imply the following conclusions:

- this algorithm seems to be an interesting alternative to another algorithms using HMM,
- it has interesting asymptotic properties,
- it can be effectively used in the combined threefold recognition algorithm, although more practical investigations should be made.

**Acknowledgement.** This work was financed by Polish Ministry of Science and Higher Education in 2005 - 2007 years as a research project No 3T11E00528.

## References

1. Brakensiek A., Rottland J., Kosmala A., Rigoll G.: Off-Line Handwriting Recognition Using Various Hybrid Modeling Techniques and Character N-Grams, Proc. of the Seventh Int. Workshop on Frontiers in Handwriting Recognition, (2000) 343-352
2. Kuncheva L.: Combining Classifiers: Soft Computing Solutions, [in.] Pattern Recognition: from Classical to Modern Approaches, Pal S., Pal A. [eds.], World Scientific (2001) 427-451
3. Marti U.V., Bunke H.: Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System, Int. Journ. of Pattern Recognition and Artificial Intelligence, Vol. 15 (2001) 65-90
4. Liu C., Nakashima K., Sako H.: Handwritten Digit Recognition: Benchmarking of State-of-the-Art Techniques. Pattern Recognition, Vol. 36. (2003) 2271-2285
5. El-Nasan A., Nagy G., Veeramachaneni S.: Handwriting recognition using position sensitive n-gram matching, Proc. 7th Int. Conf. on Document Analysis and Recognition (2003) 577-582
6. Vinciarelli A., Bengio S., Bunke H.: Offline Recognition of Unconstrained Handwritten Text Using HMMs and Statistical Language Models, IEEE Trans. on PAMI, Vol. 26 (2004) 709-720
7. Sas J., Luzyna M.: Combining Character Classifier Using Member Classifiers Assessment, Proc. of 5th Int. Conf. on Intelligent Systems Design and Applications, ISDA 2005, IEEE Press (2005) 400-405
8. Sas J.: Application of Bidirectional Probabilistic Character Language Model in Handwritten Word Recognition, Proc. of 7th Int. Conf. on Intelligent Data Engineering and Automated Learning, Springer (LNCS) (2007) 679-687
9. Kurzynski M., Zolnerek A.: Sequential pattern recognition: naive Bayes versus fuzzy relation method, Proc. of Int. Conf. on Computational Intelligence for Modelling, Control and Automation CIMCA 2005, IEEE Press, (2006) 1165-1170

---

# Semi-automatic Handwritten Word Segmentation Based on Character Width Approximation Via Maximum Likelihood Method and Regression Model

Jerzy Sas<sup>1</sup> and Marek Kurzynski<sup>2</sup>

<sup>1</sup> Technical University of Wrocław, Applied Informatics Institute  
jerzy.sas@pwr.wroc.pl

<sup>2</sup> Technical University of Wrocław, Faculty of Electronics, Chair of Systems and Computer Networks  
marek.kurzynski@pwr.wroc.pl

**Summary.** The paper presents a method of word image segmentation into images of individual characters. The method is semi-automatic, because it requires that the character sequence constituting the word on the image is known. It is assumed that widths of the characters in the alphabet are random variables and that the parameters of probability distribution are specific for each character. At the first stage of the proposed method the parameters of the distributions for all alphabet characters are estimated. Then for each word in the corpus being processed all possible segmentation variants are analyzed and for each variant its probability is calculated taking into account probability distribution of corresponding characters. Finally, such segmentation variant is selected for which the calculated probability is highest.

## 1 Introduction

There exist two general approaches to cursive handwriting recognition: "holistic" and "analytic". In holistic approach the complete word image is used as input to a word recognition algorithm. The analytic approach assumes that the word image is segmented into segments corresponding to consecutive characters and the characters are recognized individually. The analytic word recognizer applies results of character classifiers trained using the set of correctly labeled sample character images. Also some word recognizers that can be classified as holistic in fact are built of character classifiers that must be trained with isolated character samples ([1], [2]). For the process of character classifier training the set of training character samples is required. Acquisition of the training set for character recognition in cursive handwriting by thorough manual segmentation of cursive scripts is not easy and needs to be performed by an experienced individual. The training set acquisition is particularly troublesome in the case of writer dependent handwriting recognition, where the training set must be created individually for each writer. For this reason methods of automatic or semi-automatic cursive script segmentation are required in practical handwriting recognition systems.

In this paper, we consider the word segmentation method that can be classified as semi-automatic. The input to the segmentation algorithm is the set of handwritten word images. For each word image, its orthographic labeling (i.e. the sequence of letters constituting a word) is known. The aim of the method is to split the word image into the sequence of segments corresponding to the letters in the word.

The word segmentation problem defined as described above is not new. It was considered in a number of other works, but it seems that sufficiently accurate solution that could be applied in practice is still not elaborated. In [3] the authors utilize natural similarities between instances of the same letter in the text. The set of word images is segmented in such way, so as to maximize the similarities in the sets of samples of the same letter in the alphabet. Similar idea was applied in [4] but here the genetic algorithm was used to solve the problem of inter-set similarity maximization.

In [5] the estimates of letter widths measured by average stroke number per character are used. Such segmentation variant is considered as the most likely one, which minimizes the difference between actual and expected stroke number in the word image.

Most of the methods of word image segmentation are however based on prior knowledge about character shapes or geometric relations between adjacent characters ([6], [7], [8]). This knowledge is typically represented as the set of rules or as character sample sets. The application of such knowledge-based segmentation approaches for training set acquisition is limited due to great variability of writing styles and due to lack of correctly recognized character samples. Although such methods can be successfully applied at the recognition stage, they are not useful in the problem being considered in this work.

The method described in this article is an extension of the concept proposed in [5]. The width of each character appearance in the text is assumed to be a random value drawn according to a probability distribution specific for each character. The parameters of the probability distributions are estimated using the corpus of texts being segmented. Each word in the corpus is first divided into graphemes and then all segmentation variants are analyzed. Segmentation variants are obtained by grouping adjacent graphemes into subsequences representing the consecutive characters of the word. For each segmentation variant, the widths of segments hypothetically corresponding to known characters are determined, so the probability of such segmentation can be calculated using probability distribution functions of character widths and assuming the independence of character widths in the word. Finally, such segmentation variant is selected for each word in the corpus, which maximizes the calculated probability. The word and character widths can be expressed in various units. In this work, widths expressed in pixels and in graphemes were applied. Further improvement of segmentation accuracy can be obtained if segmentation assessments calculated with various width measures are combined.

## 2 Problem Statement

Let us consider the corpus  $S$  of  $N$  handwritten word images  $I_k$ ,  $k = 1, \dots, N$ , annotated with the sequence of characters actually constituting the word  $w_k = (c_k^{(1)}, c_k^{(2)}, \dots, c_k^{(l_k)})$ . Each character comes from the finite alphabet  $\mathcal{A} = (c_1, c_2, \dots, c_L)$ . The word image is normalized by applying slant and skew correction any by scaling the image so as to obtain the constant height of the horizontal base area of the handwritten word. For each word image the segmentation procedure can be applied, which divides the word image into graphemes. Graphemes are elementary graphic elements constituting the word, which can be easily extracted by relatively simple image analysis procedure. An example of correct segmentation of the word image into graphemes is presented on Fig.1. The width of the character occurrence in the word image can be characterized by its width expressed either in graphemes or in pixels.

The segmentation method presented here is based on the assumption that the character  $c \in \mathcal{A}$  in the word image has the width  $\lambda_c$ , which is observed value of a Gaussian distributed random variable  $\underline{\lambda}_c$ , with expected value  $\mu_c$  and finite-valued variance  $\sigma^2$ , equal for each  $c \in \mathcal{A}$ . Now our purpose is to estimate unknown parameters of probability distribution of random variables  $\underline{\lambda}_c$  using the known values  $d_k$  of width of the whole word  $w_k$  ( $k = 1, 2, \dots, N$ ).



Fig. 1. Exemplary word segmentation into graphemes

## 3 Estimation of Probability Distribution Parameters

In order to estimate  $\mu_c$  let us note that - under adopted assumptions - the width  $d_k$  of the whole word  $w_k$  is also observed value of random variable  $\underline{d}_k$ , which conditional expected value is given by the following regression function:

$$E(\underline{d}_k | \underline{\lambda}_c = \lambda_c) = \sum_{c \in \mathcal{A}} x_{k,c} \mu_c, \quad k = 1, 2, \dots, N, \quad (1)$$

where  $x_{c,k}$  is equal to the number of the character  $c$  occurrence in the word  $w_k$ .

Denoting by  $\epsilon$  the random element we have next:

$$d_k = \sum_{c \in \mathcal{A}} x_{k,c} \mu_c + \epsilon, \quad k = 1, 2, \dots, N, \quad (2)$$

or in equivalent compact form:

$$D = X \times M + \epsilon, \quad (3)$$

where

$$D = [d_1, d_2, \dots, d_N]^T, \tag{4}$$

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,L} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,L} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,L} \end{bmatrix}, \tag{5}$$

$$M = [\mu_1, \mu_2, \dots, \mu_L]^T. \tag{6}$$

In order to apply the mean square method to determination of regression coefficients of (3), we must minimize the following criterion:

$$Q = (D - X M)^T (D - X M), \tag{7}$$

and hence, after simple calculation we get the vector of estimates of mean values of character widths:

$$\widehat{M} = [\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_L]^T = (X^T X)^{-1} X^T D. \tag{8}$$

For estimation of common variance of characters width  $\sigma^2$  we propose to use the maximum likelihood (ML) method. Let  $w_k^{(i)} \in S$  ( $i = 1, 2, \dots, K, k = 1, 2, \dots, N_i, \sum N_i = N$ ) denotes the word containing  $i$  characters and let its width be equal to  $d_k^{(i)}$ . Applying ML method to words from  $S$ , one must first determine the joint probability density function of the word width observed values set  $T = \{d_k^{(i)}, i = 1, 2, \dots, K, n = 1, 2, \dots, N_i\}$ , which expressed as a function of unknown parameter  $\sigma^2$  is called the likelihood function. Under adopted assumptions the likelihood function can be determined as follows:

$$L_i(\sigma^2) = f(T; \sigma^2) = \prod_{i=1}^K \prod_{k=1}^{N_i} f(d_k^{(i)}; \sigma^2). \tag{9}$$

where

$$f(d_k^{(i)}; \sigma^2) = \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{i} \sigma^2} \exp\left(-\frac{(d_k^{(i)} - m_k^{(i)})^2}{2 i \sigma^2}\right) \tag{10}$$

and

$$m_k^{(i)} = \sum_{c \in w_k^{(i)}} \hat{\mu}_c. \tag{11}$$

Since the ML estimate is „the most likely“ value given the observed data, in the next step of ML procedure we maximize the likelihood function with respect to variance of characters width  $\sigma^2$ . From (10) and (11), minimizing the logarithm of likelihood function (9) we get the following estimate of characters width variance:

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^K \left[ \frac{1}{i} \sum_{k=1}^{N_i} (d_k^{(i)} - m_k^{(i)})^2 \right]. \tag{12}$$

## 4 Word Segmentation

The segmentation of words contained in the text corpus is achieved using the probability distribution of character width. Prior to the segmentation into characters the words are first segmented into graphemes. There are various methods of word image subdivision into graphemes ([1], [9]). Typically, grapheme boundaries are determined by finding characteristic points on the strokes contour. We have applied our own grapheme extraction method based on word stroke upper and lower contour analysis. In this method, grapheme boundaries are located in the areas where the vertical distance between upper and lower contour is comparable to the estimated average width of the stroke. The segmentation method is described in detail in [10]. Experiments show that character boundaries are in most cases located at the boundaries of graphemes. Correct segmentation into characters can be therefore obtained by concatenating the subsequences of adjacent graphemes. When the proposed method of grapheme extraction is applied, the number of graphemes per single character very rarely exceeds 4. Let us consider the word  $w_k$  consisting of  $l_k$  characters. Let the number of graphemes for this word be  $m_k$ . All possible segmentation variants of the word  $w_k$  can be obtained by creating all possible subdivisions of grapheme sequence into  $l_k$  subsequences. Let  $V_n^k$  denote  $n$ -th segmentation variant for the word  $w_k$ . By  $d_j^k(n)$  we denote the length of the grapheme subsequence corresponding to  $j$ -th character in the word. By assuming the independence of random variables representing actual widths of characters in the word, the probability of the segmentation variant can be calculated as:

$$p(V_n^k) = \prod_{j=1}^{l_k} f(d_j^k(n); \mu_{c_j^k}, \sigma^2), \quad (13)$$

where  $f(x; \mu_{c_j^k}, \sigma^2)$  is the probability distribution function for  $j$ -th character in the word  $w_k$ . The length of the word segment corresponding to  $j$ -th character can be expressed in various units. In our experiments we applied widths measured directly by graphemes count per segment and by the number of pixel columns covering all strokes of the segment. For both measures, corresponding sets of probability distribution parameters must be calculated according to formulas (8) and (12). In result for each segmentation variant, two probabilities for grapheme-based and pixel-based widths can be calculated:  $p^G(V_n^k)$  and  $p^P(V_n^k)$ .

The probabilities calculated in this way can be treated as likelihoods of segmentation variants permissible for the word. In many other works combination of likelihoods evaluated using various presumptions leads to the improvement of the final decision relevance. The previous experiences in applying various simple combination formulas to similar problems show that the best results are usually attainable if multiplicative combination of likelihoods is applied ([10], [11]). Final likelihood assessment of the segmentation variant is therefore evaluated as:

$$I(V_n^k) = p^G(V_n^k)p^P(V_n^k) \quad (14)$$

Finally, such segmentation variant  $V_{n^*}^k$  is selected which maximizes its likelihood assessment  $I(V_{n^*}^k)$ . The complete text corpus segmentation algorithm consists of the following steps:

```

split each word image into graphemes;
calculate the parameters of pdf functions for all characters
  in the alphabet and for widths expressed in graphemes
  and in pixels;
for each word in the corpus:
{
  create the set of segmentation variants taking into account
    the limits of grapheme count per character;
  for each segmentation variant:
    calculate the likelihood assessment;
  select this segmentation variant
    for which the calculated assessment is highest;
}

```

## 5 Experimental Results

The aim of experiments carried out was to evaluate the word segmentation accuracy achieved by the methods described here. In particular, the comparison of the segmentation accuracy of simple and combined methods has been carried out.

The segmentation methods were tested using the set of word images written by three writers. The writers participating in the experiment were asked to rewrite the same text samples. Because the work described here is a part of a wider project related to the problem of handwritten medical documents recognition, the texts used in the experiment were extracted from authentic medical reports written in Polish. Only words not including Polish diacritic characters were used. The experiment participants were selected so as to obtain the writing style samples of different precision and readability. Obtained writing styles can be categorized as: calligraphic, fair and chaotic. The corpus of word images used in the experiment consists of 216 words containing total 1355 character appearances.

The performance of semi-automatic words segmentation was evaluated by individually assessing each segment corresponding to the known character in the word image. Each segment was evaluated by a human and categorized as correct or incorrect. The criterion of the categorization was the readability of the corresponding character shape in the extracted fragment of the image. Final accuracy of the segmentation for each writer was evaluated as a fraction of correctly assessed segments in total number of character appearances in the test data set. The segmentation accuracies obtained by simple methods based on single width measures and by the combined method for the three writing styles are given in Table 1.



**Table 1.** Word segmentation accuracies achieved by proposed algorithms

Method	Calligraphic	Fair	Chaotic
grapheme-based	0.78	0.73	0.66
pixel-based	0.85	0.72	0.71
combined	0.91	0.84	0.76

The accuracy of word segmentation is highest for calligraphic writing style for all methods. It is probably caused by the fact that the repeatability of character shapes in neat writing is higher, so the differences between widths of various instances of the same character is lower than in the case of more chaotic writing style. In the case of all writing styles the accuracy of the combined method is observably higher than the accuracy of the better of component simple methods.

## 6 Conclusions

The results of semi-automatic segmentation obtained with the proposed methods seems to be practically applicable in the case of neat writing styles where the character widths are highly repeatable. In the case of more chaotic writing styles the resultant segmentation needs further improvements. It can be probably achieved by applying slight changes in positioning of segment boundaries so as to increase to inter-set similarities of samples gathered for the same alphabet characters.

The main methodological weakness of the method is the assumption of the equality of probability distribution variances for various letters. Better results can be probably obtained if the variances are estimated individually for each character.

**Acknowledgement.** This work was financed from the Polish Ministry of Science and Higher Education resources in 2007-2009 years as a research project No 3 T11E 005 28.

## References

1. Arica, N., Yarman-Vural, F.: Optical character recognition for cursive handwriting. *IEEE Trans. on PAMI* **24** (2002) 801–813
2. Marti, U., Bunke, H.: Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *Int. Journ. of Pattern Recognition and Artificial Intelligence* **15** (2001) 65–90
3. Schomaker, L., Teulings, H.: Unsupervised learning of prototype allographs in cursive script using invariant handwriting features. In J.C. Simon, S.I., ed.: *From Pixels to Features III*, Amsterdam, North-Holland (1992)

4. Sas, J., Markowska-Kaczmar, U.: Semi-automatic training sets acquisition for handwriting recognition. In Saed, K., ed.: Proc. of 6th Int. Conference on Computer Information Systems and Industrial Management Applications - CISIM 2007, IEEE Press (2007)
5. Mackowiak, J., Schomaker, L., Vuurpijl, L.: Semi-automatic determination of allograph duration and position in on-line handwriting words based on the expected number of strokes. In: Progress in Handwriting Recognition, London: World Scientific (1997)
6. Pal, U., Belaid, A., Choisy, C.: Touching numeral segmentation using water reservoir concept. Pattern Recognition Letters (2003) 261–272
7. Xiao, X., Leedham, G.: Knowledge based english cursive script segmentation. Pattern Recognition Letters (2000) 945–954
8. Sadri, J., Suen, C., Bui, T.D.: A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings. Pattern Recognition **40** (2007) 898–919
9. El-Yacoubi, A., Gilloux, M., Sabourin, R., C.Y., S.: An hmm-based approach for off-line unconstrained handwritten word modeling and recognition. Int. Journ. of Pattern Recognition and Artificial Intelligence **21** (1999)
10. Sas, J.: Combined approach to semi-automatic handwritten word segmentation based on character width approximation. In: Proc. of XVI Int. Conference on Systems Science. (2007) – to be published
11. Sas, J.: Application of bidirectional probabilistic character language model in handwritten words recognition. In E. Corchado, H. Yin, V.B.C.F., ed.: Intelligent Data Engineering and Automated Learning - IDEAL 2006, Springer Verlag, Berlin, Heidelberg (2007)

---

# Sentence Boundary Verification in Polish Text

Krzysztof Simiński

Institute of Informatics, Silesian University of Technology  
Krzysztof.Siminski@polsl.pl

**Summary.** In this paper the heuristic method based on phrase analysis is proposed for sentence boundary verification in Polish texts. The decision rules, maximum entropy and neural network as reference methods are compared with the phrase analysis. The results elaborated by the proposed method are more accurate than the reference methods.

## 1 Introduction

The sentence boundary problem splits into two tasks. The first one is the sentence boundary detection in texts with no punctuation marks and no upper and lower case discrimination. The latter one is the sentence boundary verification in texts with punctuation marks and preserved letter cases.

This paper deals with the verification of sentence boundaries in Polish texts.

The problem of proper sentence boundary verification is crucial in natural language processing. The analysis is mostly based on the sentences. The danger of the vicious circle can appear: for proper analysis the text should be split into sentences, but the analysis should be carried out to split the text into sentences. Polish is an inflecting language and thanks to this many steps of analysis can be based on the morphological forms of words.

The sentence is not easy to define. There are more than two hundred definitions of sentence. The most popular belong to one of three groups:

1. The formal definition. Sentence is a text fragment that starts with capitalised word and ends with the full stop (or other punctuation mark that denotes the end of sentence).
2. The quantum of information. The sentence is a text fragment containing the separate autonomous quantum of information.
3. The linguistic definition. The sentence is a set of words joined by predicative syntactic determinance [10].

The problem arises as the punctuation marks (mainly the full stop) have many functions. In Polish the full stop denotes the end of sentence, the abbreviation, the ordinal numeral, thousands separator in numerals (cf. table 1).

The additional problem that complicates the task of sentence boundary verification is the haplography. It occurs when the letter or sign is written only once when it should be written twice. The full stop being the element of the phrase

**Table 1.** The multiple function of full stop in Polish

function	Polish example	English translation
end of sentence	To jest dom.	This is a house.
abbreviation	prof. Nowak	prof. Nowak
omission	Widział (...) psa.	He saw (...) a dog.
ordinal numeral	w 5. domu	in the 5th house
thousand separator	1.234,56	1,234.56

at the end of the sentence merges with the full stop denoting the end of sentence into singular punctuation mark, eg. *Wiem, co powiedział pan A.* ‘I know what Mr A. said.’ The other punctuation marks do not manifest the haplography phenomenon, eg. *Co powiedział pan A.?* ‘What did Mr A. say?’

## 2 Methods

There are very few papers on verification of sentence boundaries in Polish text. The methods used in other languages can be grouped into the direct analysis, the analysis of characters, the analysis of features.

### 2.1 The Direct Analysis

The direct model is based on implementation of situations when the full stop does not denote the end of sentence, eg. the full stop before the subsequent question mark does not separate the sentences.

This method implies the strict collaboration of computer and linguistic specialists. The satisfactory results can be achieved with this method. The new cases need the interference in the programme code, so the maintainance is expensive and prone to errors [11].

### 2.2 The Analysis of Characters

The regular expression are used to find cases when the full stop does not denote the sentence boundary. The common cases are the WWW addresses, dates in yyyy.mm.dd format or numbers [1, 11].

### 2.3 The Analysis of Features

Most common method for the sentence boundary verification is the analysis of features. For a lexem that can denote the end of sentence (ie. ‘.’, ‘?’, ‘!’) the features of this lexem, the preceding and following lexems can be extracted. On these features the attempts are conducted to determine the function of the full stop, question mark or exclamation mark (whether they denote the end of sentence or not).

The lexems the features are extracted from constitute the feature window (called also a context). Most commonly the window is symmetrical and contains a few lexems before the punctuation mark in question and a few after (3-element [3, 11], 5-element [4], 7-element [8, 4], 9-element window [4]).

The decision  $d$  whether the full stop ends the sentence is an element of the set  $\mathcal{D} = \{\text{end}, \text{no end}\}$ . The values of features (the attributes)  $a$  in a window build up the feature vector  $\mathbf{A} = \{a_1, a_2, \dots, a_n\}$ . All vectors are gathered in the set  $\mathcal{A} = \{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_m\}$ . From the training texts all pairs  $\langle \mathbf{A}, d \rangle$  are extracted. In the test texts for each vector  $\mathbf{A}$  the decision  $d$  about the function of the punctuation mark is unknown and should be determined.

From the training texts the features are extracted for each occurrence of full stop, question mark and exclamation mark. The extracted features vary in papers, the most common are: the length of lexem, the part of speech it represents, whether the lexem represents the numerical value, whether it is a punctuation mark (full stop, question mark, exclamation mark, quote, colon, semicolon, ...), whether it is uppercase, whether it is an abbreviation, whether the lexem is honorific (eg. *Mr.*, *Dr.*, *Gen.*) [3, 11, 7].

The function of the full stop, question mark or exclamation mark in the sentence is predicted by means of decision trees [4], decision rules [8], hidden Markov models [12], maximum entropy [6] and artificial neural networks [4].

### 3 Our Approach

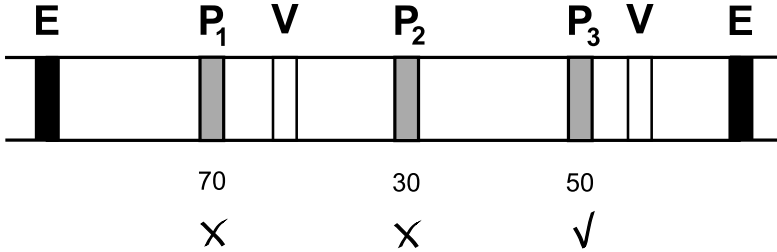
One can ask whether verification of the sentence boundaries can be simplified by using more intuitive methods based on the linguistic approach. In some phrases the full stop is an element of phrase and does not denote the end of sentence. It is advantageous to detect the phrases where the full stop is an element of the phrase. Some phrases never occur at the end of the sentence, eg.: *prof.* ‘prof.’, *np.* ‘eg.’, *św.* ‘holy’. So the full stop in these phrases is never the indication of the sentence boundary.

There are phrases with the full stop that may occur in the sentence or at the end of the sentence, eg.: *itd.* ‘etc’, *p. n. e.* ‘BC’. In order to determine whether the phrase ends the sentence or not the following heuristic algorithm has been proposed.

The assumptions of the algorithm are following (1) the sentence is a simple one (has one verb) and (2) each phrase has the end weight  $w_e \in [0, 100]$ . End weight represents the probability that the phrase ends the sentence. The phrases that never end the sentence have  $w_e = 0$  and those that always end the clause have  $w_e = 100$ .

The figure 1 depicts the algorithm. First the certain sentence boundaries are found (marked in the figure 1 with letter **E**). These are the situations when without doubt the punctuation marks denote the ends of sentences. The remaining phrases have  $w_e < 100$  and should be carefully treated. All the phrases with  $w_e < 100$  that appear between the certain sentence boundary **E** and the verb **V** do not denote the sentence boundary with no regard to their end weights

(if the phrases denoted the sentence boundaries some sentences would have no verb). From the phrases between verbs the one with the maximum end weight is selected to denote the sentence boundary. In the figure 1 the phrase  $P_1$  is not a sentence boundary because it is between certain sentence boundary and the verb. From the phrases  $P_2$  and  $P_3$  between two verbs the phrase  $P_3$  has maximum end weight  $w_e = 50$  and denotes the sentence boundary.



**Fig. 1.** The symbolic excerpt of the text with phrases and verbs. **E** – certain end of sentence, **P** – phrase, **V** – verb. Numbers below the phrases are the their end weights  $w_e$ . The phrase  $P_1$  does not denote the sentence boundary because it is between the certain end of previous sentence and the verb. From the phrases  $P_2$  and  $P_3$  (that are between verbs) the phrase with maximum end weight is selected. In this figure only the phrase  $P_3$  represents the end of the sentence.

### 4 Experiments

The are very few papers on the sentence boundaries verification in Polish texts. In order to compare the results elaborated by the analysis of phrases methods used in other languages have been used as the reference methods. The decision rules, maximum entropy and neural network have been applied to determine the baseline for the research.

The analysis has been conducted on 128 Polish newspaper texts (containing 114 635 words). The newspaper texts have been used because their contain more situations of sentence boundaries than the *belles-lettres*. The set of texts has been split into traning and test sets. The advanced morphological analysis by means of LAS (Advanced Linguistic Server developed in the Institute of Informatics at the Silesian University of Technology [9, 2]) has been applied to these texts. From 44 training texts the 2198 tuples of features have been extracted. The window for feature extraction has comprised 5, 7 or 9 elements. The feature for the punctuation mark in question is enumerative: whether it is ‘.’, ‘?’ or ‘!’. The extracted features for the remaining lexems in the window are following: the length, is capitalised?, is a numeral?, is a Roman numeral?, is known word?, is a punctuation mark?

The decision tree and subsequently the rules has been induced with C4.5 algorithm [5].

The network used in the system was the tree layer network with one input neuron for each feature and one output neuron. The activation function for all the neurons was the sigmoid function. The weights of the neurons were adjusted by means of back propagation.

## 5 Results

The results are presented in the table 2. The *situation* is a full stop, exclamation mark or question mark. The correctness measure  $p$  is defined as

$$p = \frac{\text{well predicted situations}}{\text{all situations}}$$

The reference correctness  $p_r$  is the correctness when all situations are regarded as the ends of sentences. The reference correctness in test texts is  $p_r = 81.2\%$ .

The table 2 presents the average correctness for test texts elaborated with different methods and window width. The results for the most effective window in each reference method are shown in the table.

**Table 2.** The best results of sentence boundary verification

method	correctness [%]	window width
rules	95.1	5, 9
maxent	93.9	9
neural network	94.8	5
phrases	97.1	–
reference correctness	81.2	

The table 3 shows the comparison with sentence boundary verification in other languages.

**Table 3.** The comparison with other languages (en – English, fr - French, ge – German, gr – Greek)

method	correctness [%]		
language	Polish	other	
rules	95.1	76.9 (en), 97.7 – 99.3 (ge), [12, 4]	99.4 (fr, gr) [4, 8]
maxent	93.4	97.6 (en), 97.9 – 98.8 (en) [12, 7]	
neural network	94.8	98.7 (ge), 99.4 (fr)	[4]
hidden Markov model		94.5 (en)	[12]
phrases	97.1		

The experiments with large window reveal that for the best result the window should comprise only a few elements [4].

## 5.1 Problems

The method has not 100% correctness. The analysis of errors shows the sources of uncorrect sentence boundary denotation:

1. the internet addresses: *www.polsl.pl*;
2. dates: *2007.05.16*;
3. the ambiguity of phrases – the phrase is both an abbreviation and a valid word, eg.: *ul.* – abbreviation of word *ulica* ‘street’ but *ul* is also a valid word (‘hive’): *Na ul. Miodowej stoi ul.* ‘In the Honey Street [there] is a hive.’;
4. the ambiguity of morphological forms – the form of a verb (that is crucial in phrase analysis) is the same as the form of another part of speech, eg. *pchła* – ‘a flea’, *pchła* – ‘[she] pushed’;
5. the compound and complex sentences – the assumption of the method is that each sentence has one verb (is a simple sentence), so the method does not always (but surprisingly often) treat the compound and complex sentences properly;
6. foreign word phrases.

## 6 Conclusions

The phrase analysis in sentence boundary verification in Polish text is better than reference methods of features analysis. The phrase analysis is simple and fast (especially when compared with the learning procedures of neural network). The method does not require the voluminous phrase list (147 phrases).

The phrase analysis is vulnerable to unknown and foreign language phrases. The method is not always satisfactorily effective with compound sentences.

The method offers sufficiently accurate sentence boundary verification for Polish text basing only on the morphological analysis. The improvement requires syntactic analysis (compound and complex sentences, disambiguation of morphological forms).

## References

1. Krzysztof Jassem. *Przetwarzanie tekstów polskich w systemie tłumaczenia automatycznego POLENG*. Wydawnictwo Naukowe UAM, 2006.
2. Sławomir Kulików. Implementacja serwera analizy lingwistycznej dla systemu THETOS – translatora tekstu na język migowy. *Studia Informatica*, 24(3 (55)):171–178, 2003.
3. Kelley Herndon Ford Neha Agarwal and Max Shneider. Sentence boundary detection using a maxent classifier.
4. David D. Palmer and Marti A. Hearst. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23(2):241–267, 1997.



5. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.
6. Adwait Ratnaparkhi. A simple introduction to maximum entropy models for natural language processing. Technical report, Institute for Research in Cognitive Science, University of Pennsylvania, 1997.
7. Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19.
8. E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic extraction of rules for sentence boundary disambiguation. *Proceedings of the Workshop in Machine Learning in Human Language Technology, Advance Course on Artificial Intelligence*, pages 88–92, 1999.
9. Nina Suszczańska, Mirosław Forczek, and Artur Migas. Wieloetapowy analizator morfologiczny. *Speech and Language Technology*, 4:155–165, 2000.
10. Stanisław Urbańczyk, editor. *Encyklopedia języka polskiego*. Zakład Narodowy im. Ossolińskich – Wydawnictwo, 1991.
11. Daniel J. Walker, David E. Clements, Maki Darwin, and Jan W. Amtrup. Sentence boundary detection: A comparison of paradigms for improving MT quality. In *MT Summit Proceedings VIII*, September 2001.
12. Haoyi Wang and Yang Huang. Bondex – a sentence boundary detector, 2003.

**Medical Applications**

---

# Synthesis of Static Medical Images – An Example of Melanocytic Skin Lesions

Zdzisław S. Hippe<sup>1</sup> and Łukasz Piątek<sup>2</sup>

<sup>1</sup> Department of Expert Systems and Artificial Intelligence, University of Information Technology and Management, 35-225 Rzeszów, Poland

`zshippe@wsiz.rzeszow.pl`

<sup>2</sup> Department of Distributed Systems, University of Information Technology and Management, 35-225 Rzeszów, Poland

`lpiatek@wsiz.rzeszow.pl`

**Summary.** Some algorithms to synthesise static images of melanocytic skin lesions are briefly outlined. The key question of the compiled synthesis methodology is a semantic conversion of textual description of melanocytic skin lesions by an in-house developed algorithms capable to create hybrid (vector-raster) images. It was assumed, that owing to implementation of randomized development of images, single textual data vector enables creation a set of images, corresponding to symptoms of real melanocytic skin lesions. It was found, that the developed methodology of synthesis images of the lesions being investigated can be successfully used in the process of teaching of dermatology students and also in training of preferred medical doctors.

## 1 Introduction

In the past few years an increasing interest in images of melanocytic skin lesions is observed, what could be treated as a visual support of diagnosis of malignant melanoma, currently one of the most dangerous type of cancers [1]. Additionally, the lack of professional computer databases containing images of above mentioned lesions is clearly noticed. This situation, at least in this country, is probably the result of various difficulties in the development such databases, among them by a personal data protection act. Present interpretation of this act imposes the necessity to obtain patient's approval not only for making the picture of real lesion in hospital or clinic, but also the permission for publishing it or even handing it over to another scientific research institution, that specializes in processing digital images. These reasons inspired us to start some research on the development of generalized algorithms of synthesis of medical images in general, but specifically on synthesis of static images of melanocytic skin lesions, and then in implementation of these algorithms in a computer program system, allowing to create multi-category informational database, containing both synthesized and real images of skin lesions.

## 2 Methodology of the Research

According to our preliminary results of the synthesis of skin lesion images [2], the semantic conversion of records from textual informational database [3] into respective digital images [4], seemed to be quite reasonable way to achieve the main goal of our research. The source (textual) database available in our group, contains information of roughly **550** real cases of anonymous patients' lesions, confirmed by hisopatological examinations. Each case is described according to the methodology of analysis of real melanocytic skin lesions established some time ago by Braun-Falco [5], recently improved by Stolz [6]. The analytical methodology relies on application of the so called **ABCD** rule, in which **A** (*Asymmetry*) means result of evaluation of lesion's asymmetry, **B** - (*Border*) the result of estimation of character of the rim of the lesion, **C** - (*Color*) identifies number of colors present in the lesion (out of 6), and **D** - (*Diversity of structures*) stands for the number of structures (out of 5). Elements of **ABCD** rule enumerate four main symptoms of an investigated lesion, and at the same time these elements are used to compute the **TDS** (**T**otal **D**ermatoscopy **S**core) parameter. According to its value, the analyzed lesion could be classified to one of four accepted categories (classes) of melanocytic skin lesions, namely: *Benign nevus*, *Blue nevus*, *Suspicious nevus* or *Melanoma malignant*. The value of the **TDS**-parameter, accordingly to [5], is computed using the following formula:

$$TDS = 1,3 * \mathbf{A}symmetry + 0,1 * \mathbf{B}order + 0,5 * \sum \mathbf{C}olor + 0,5 * \sum \mathbf{S}tructure \quad (1)$$

According to this equation (1), the highest coefficient is assigned to the symptom *asymmetry*, what implicates the necessity of forming particularly precised algorithm of synthesis of this symptom. On the other hand, logical values of coefficients rated to *Color* and *Diversity*, are dependent only on number of separate features of a given type (*color* and *structure*), without diversifying their influence on the character of the lesion. The first results of our previous research [2], showed that the transformation of a single case from our informational database allowed to obtain only one synthetic image. In a new approach described here, a random inclusion of particular parts of created lesion's image (we are talking here about *asymmetry* and *border* of a lesion), allows to generate a number of synonymous images [7], i.e. images characterized by the possession of selected symptoms containing the same logical values. Therefore, for 548 available textual vectors, it is possible to obtain several thousand of synthesised digital images.

## 3 Algorithms of Mapping of Lesions' Symptoms

Algorithms developed by us are significantly different from those already described in [4]. The first difference is the use of hybrid type (vector-raster) approach to synthesis of images. Vector technique is applied for the synthesis of

lesion's asymmetry (mapping the value of the  $\langle \mathbf{Asymmetry} \rangle$  attribute), through the use of **NURBS**, i.e. **non-uniform rational B-spline** [8], whereas raster operations are implemented for mapping of remaining symptoms of lesions. The latter function is realised by putting on so called bitmaps, used to represent 2-dimensional graphics in the form of 2-dimensional matrix, whose elements determine location of any single pixel of image and its color in RGB color mode. The second, entirely new approach to gain synthesis of skin lesion images is the inclusion the latest results of the research executed at Kansas University on meaning of lesion's asymmetry [9, 10]. Additionally, these results differentiate a role of particular colors and structures, within a lesion, allowing to determine the logical value of a new total dermatoscopy score parameter, **New\_TDS** (see equation (2)).

$$\begin{aligned}
 TDS = & (0,8 * Asymmetry) + (0,11 * Border) + (0,5 * C\_White) \\
 & + (0,8 * C\_Blue) + (0,5 * C\_DarkBrown) \\
 & + (0,6 * C\_LightBrown) + (0,5 * C\_Black) \\
 & + (0,5 * C\_Red) + (0,5 * Pigment\_Network) \\
 & + (0,5 * Pigment\_Dots) + (0,6 * Pigment\_Globules) \\
 & + (0,6 * Branched\_Streaks) + (0,6 * Structureless\_Areas)
 \end{aligned} \tag{2}$$

In this paper only algorithms of mapping of *asymmetry* (see Sect. 4) and lesion's *border* (see Sect. 5) are presented. However, to perform full experiment, in all presented here images, instead of symptoms  $\langle \mathbf{Color} \rangle$  and  $\langle \mathbf{Diversity\ of\ structure} \rangle$  a special feature (called by us "standard background") was added.

## 4 Algorithm of Synthesis Lesion's Asymmetry

Synthesis of lesion's asymmetry consists of using non-uniform rational B-spline (i.e. NURBS surfaces - see Fig. 1), which could be treated as more generalized form of Bezier's curved and surfaces. NURBS surface is outlined on the basis of two parameters  $u = (0,1)$  and  $v = (0,1)$ , and four variables  $x(s,t)$ ,  $y(s,t)$ ,  $z(s,t)$  and  $d(s,t)$ , so it could be represented as following equation:

$$\mathbf{S}(u, v) = \left( \frac{x(u, v)}{d(u, v)}, \frac{y(u, v)}{d(u, v)}, \frac{z(u, v)}{d(u, v)} \right) \tag{3}$$

The developed methodology enable to obtain 3-dimensional smooth surfaces having complicated shapes, that facilitate mapping of lesion's asymmetry, and also to add the 3rd dimension, useful in selecting lesion's categories (mainly for the *Malignant Melanoma*). The process of mapping the type of image's asymmetry requires obtaining specific shapes of outline edge of defined surface (Fig. 2), through defining coordinates of control points, located on the border of surface. Depending on the value of  $\langle \mathbf{Asymmetry} \rangle$  attribute there are 3 possible procedures:

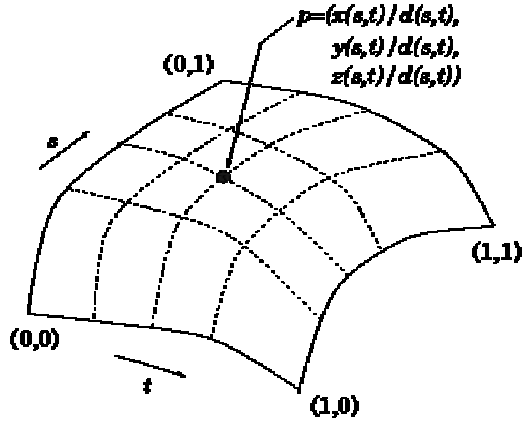


Fig. 1. NURBS surface

- for *<symmetric lesion>* all points placed on the lesion's border should be symmetric according to 2-fold symmetric axes (Fig. 2a),
- for *<one-axial asymmetry lesion>* only one 2-fold symmetric axis is required (Fig. 2b),
- for *<two-axial asymmetry lesion>* object cannot have any 2-fold symmetric axis (Fig. 2c).

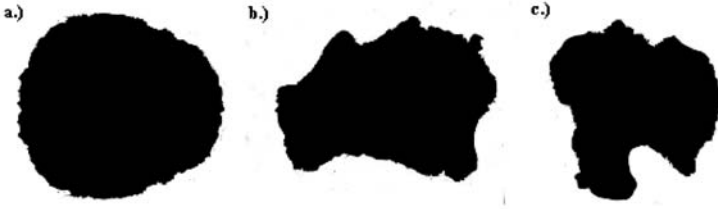
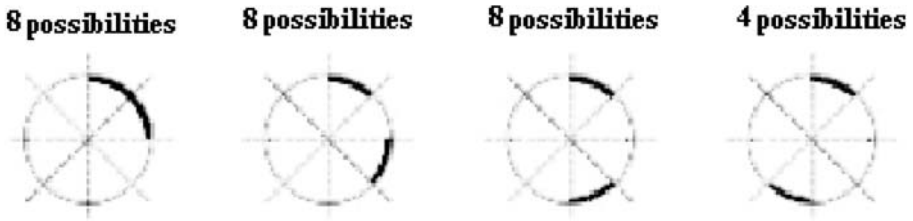


Fig. 2. Examples of NURBS surfaces defined as symmetric lesion (a), one-axial asymmetry lesion (b) or two-axial asymmetry lesion (c)

## 5 Algorithm of Synthesis of Border of the Lesion

The definition of the character of a rim of lesion, then for each of 8 equal parts of the synthesised image, relies on splitting it by 4 axes crossed in a point. We need to define sharpness of transition between a lesion and healthy skin. This transition is **sharp** (adding value 1 to the score) or **diffuse** (without changing the score). Depending on the value of *<Border>* attribute obtained in this particular way, there are **256** possible combinations of sharp/diffused transitions [4]. For example, for the *<Border>* equals to **2**, we can have **28** synonymous images (Fig. 3). The diffusion of selected octal parts of an image is accomplished by



**Fig. 3.** Collection of 28 possible sharp/diffuse combinations of fragments. Sharp transition from a lesion towards the skin is displayed here by means of a thick line, whereas thin line represents "fuzzy" conversion. Each of 4 presented cases is multiplied by 4 or 8, because a set of 4 or 8 new transitions can be generated applying the operation of four- or eight-fold symmetry axis, perpendicular to the plane of an image.

means of specific conversion based on low-capacity Gaussian filter [11] for the regarded fragments.

## 6 Program Implementation

Developed algorithms of synthesis of lesions' images are implemented in **C++** language, combined with the use of **MFC** (**M**icrosoft **F**oundation **C**lasses) library [12] as well as **OpenGL** graphic library [13] (using pre-defined functions to generating **NURBS** surfaces, putting on bitmaps and geometrical transformations).

## 7 Experiments

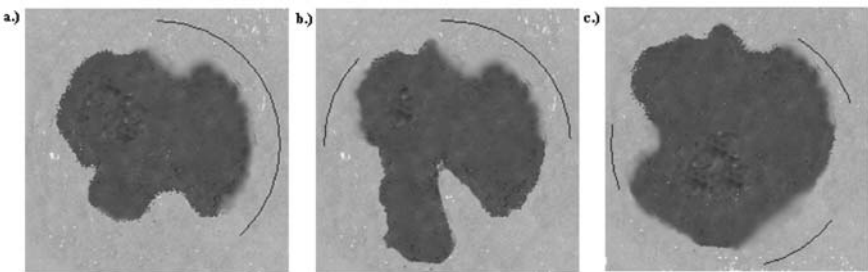
Real cases of lesions description, using the attributional logic, were in the form of 15-element vectors. Logical values of descriptive attributes allowed are gathered in Tab. 1. Thanks to implementing randomized mapping of lesions *asymmetry* and *border* in developed algorithms, it was possible to get a set of synonymous images (see Fig. 4) for a single textual data vector. Comparing Fig. 4a - 4c it can be noticed that each of synthesised images is characterized by different shape (mapping  $\langle \mathbf{A}symmetry \rangle$  attribute), and various combinations of fragments of the rim.

## 8 Summary and Conclusion

In the research described here, we succeeded in obtaining required random synthesis of selected symptoms of real melanocytic skin lesions, namely:  $\langle \mathbf{A}symmetry \rangle$  and  $\langle \mathbf{B}order \rangle$ . The developed algorithms enables to generate the exhaustive number of synthesized images, corresponding to symptoms contained in a given lesion, originally described by textual vector from

**Table 1.** 15-component textual data-vector

Descriptive attribute	Logical values
<Asymmetry>	<symmetric lesion> <one-axial asymmetry lesion> <two-axial asymmetry lesion>
<Border>	Even numerical value oscillating between <b>0</b> and <b>8</b>
<Color>	
- black	present / absent
- blue	present / absent
- dark-brown	present / absent
- light-brown	present / absent
- red	present / absent
- white	present / absent
<Diversity of structures>	
- branched streaks	present / absent
- pigment dots	present / absent
- pigment globules	present / absent
- pigment network	present / absent
- structureless area	present / absent
<b>TDS (Total Dermatoscopy Score)</b>	Numerical value computed according to the equation (1)
Lesion's category	<benign nevus> <blue nevus> <suspicious nevus> <malignant melanoma>



**Fig. 4.** Example set of synonymous images of lesion. Diffusion of selected octal parts of image are marked by the black curves.

the source database. Synthesized images enable to create multi-category informational database, which can be successfully used not only in teaching of medicine students, but also in job practice of dermatologists and preferred medical



doctors. It seems that synthetic images, generated by means of the developed algorithms, can be successfully used as an alternative source of knowledge in relation to digital medical images. In further research, the problem of correct synthesis other symptoms of melanocytic skin lesions, namely *<Color>* and *<Diversity of structure>* will be dealt with.

## References

1. Stolz W., Braun-Falco O., Bilek P., Landthaler M., Burgdorf W. H. C., Cognetta A. B. (2006) The Atlas of dermatology. In: Czelej D., Michalska-Jakubus M., Ziarkiewicz M., Czelej Sp. z o.o., Lublin (in Polish).
2. Hippe Z.S., Piątek Ł. (25-27.09.2005) From research on the database of simulated medical images. Proc. The Conference Databases for Science and Technology, Gdańsk, 225–230 (in Polish)
3. Hippe Z.S. (1999) Computer Database 'NEVI' on Endangerment by Melanoma. TASK Quarterly 1999(3,No4)483-488
4. Hippe Z.S., Grzymała-Busse J.W., Piątek Ł. (2006) Randomized dynamic generation of selected melanocytic skin lesions features. In: Kłopotek M.A., Wierzchoń S.T., Trojanowski K. (Eds.) Advances in Soft Computing (Intelligent Information Processing and Web Mining), Springer-Verlag, Berlin-Heidelberg, 21-29
5. Braun-Falco O., Stolz W., Bilek P., Merkle T., Landthaler M. (1990) Das dermatoskop. Eine Vereinfachung der Auflichtmikroskopie von pigmentierten Hautveränderungen. Hautarzt 1990(40)131
6. <http://www.pless.fr/dermatoscopie/>
7. Kulikowski J.L. (25-27.09.2005) The foundations of the structural description of distracted databases of expert knowledge. Proc. The Conference Databases for Science and Technology, Gdańsk, 29–38 (in Polish)
8. Angel E. (2000) Interactive Computer Graphics - A Top Down Approach with OpenGL. Addison Wesley Longman Inc., 445-446
9. Alvarez A., Bajcar S., Brown F.M., Grzymała-Busse J.W., Hippe Z.S. (2003) Optimization of the ABCD Formula Used for Melanoma Diagnosis In: Kłopotek M.A., Wierzchoń S.T., Trojanowski K. (Eds.), Advances in Soft Computing, (Intelligent Information Processing and Web Mining), Springer-Verlag, Heidelberg, 233-240
10. Grzymała-Busse J.W., Hippe Z.S. (2005) Data Mining Methods Supporting Diagnosis of Melanoma. Proc. of the 18th IEEE Symposium on Computer-Based Medical Systems, IEEE Comp. Soc., Los Alamitos (California, USA), 371-373.
11. Lifshitz L.M., Pizer S.M. (1990) A multiresolution hierarchical approach to image segmentation based on intensity extrema. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, 529-540
12. <http://www.mfc.org.pl/>
13. <http://www.opengl.org/>

---

# Morphological Spectra as Tools for Texture Analysis

Juliusz L. Kulikowski<sup>1</sup>, Malgorzata Przytulska<sup>2</sup>, and Diana Wierzbicka<sup>3</sup>

Institute of Biocybernetics and Biomedical Engineering Polish Academy of Sciences,  
Ks. Trojdena Str.4 02-109 Warsaw, Poland  
{jlkulik, gosia, diana}@ibib.waw.pl

**Summary.** There are presented formal properties of morphological spectra as a novel tool for analysis of textures. It is described a multi-level structure of the system of morphological spectra and the method of calculation of spectral components. Formal properties of morphological spectra: symmetries, ability to describe parallel shifts and rotations of analyzed images are presented. Several comments concerning practical aspects of using morphological spectra to analysis of textures are also given.

## 1 Introduction

Analysis of textures is widely used in numerous areas of image processing and pattern recognition. It may play a role of an autonomous task (say, in classification of specimens of building materials) as well as it may be a step to a selection of regions of interest (ROI) in examination of biomedical images, aerial photos, etc. Numerous methods based on morphological, statistical, fractal and many other approaches for this purpose have been proposed [1][4][6][6]. However, typical texture analysis methods: morphological, statistical, etc. are rather ineffective in description of multi-level textures, spectral and fractal methods, due to their multi-scalar organization, suit much better to it. In [2] a method of texture analysis based on a set of multi-level image transformations has been proposed. The method then has been improved and simplified due to a concept of multi-level morphological spectra reached in sequences of standard linear operations and permutations [3]. In this paper the method is only shortly presented, attention being mostly focused on some formal properties of morphological spectra as a tool for texture analysis. The organization of the paper is as follows: in Sec. 2 basic concepts of morphological spectra are presented. In Sec. 3 some properties of symmetry of morphological spectra and of their invariance with respect to parallel image shifts and rotations are described. Several remarks concerning using morphological spectra to analysis of textures have been described in Sec. 4. Conclusions are given in Sec. 5.

## 2 Basic Concepts of Morphological Spectra

There are considered 2D monochromatic discrete images given in the form of arrays:

$$\mathbf{u} = [u_{i,j}], \quad i \in [1, 2, \dots, I], \quad j \in [1, 2, \dots, J], \quad (1)$$

where  $I$  and  $J$  are integers denoting, respectively, the number of rows and columns,  $u_{i,j} \in [0, 1, \dots, 2^N]$  are integers representing local luminance intensities. The triples  $[i, j, u_{i,j}]$ , called pixels, are basic elements of the image. It will be assumed that  $I \leq 2^m$  and  $J \leq 2^m$ ,  $m$  being minimal integer  $> 0$ , satisfying both inequalities. In the case of  $I < 2^m$  or  $J < 2^m$  the lacking pixel values can be filled by zeros, the given pixels  $u_{i,j}$  being located as close as possible to the upper-left corner of the  $2^m \times 2^m$  array.

Morphological spectra can be defined as a hierarchy of arrays obtained by sequences of transformations performed on  $\mathbf{u}$  as follows.

The original image  $\mathbf{u}$  is defined as a 0-th level morphological spectrum,  $\mathbf{u} \equiv \mathbf{S}^{(0)}$ .

The 1-st level morphological spectrum  $\mathbf{S}^{(1)}$  consists of four arrays of  $2^{m-1} \times 2^{m-1}$  size, called sub-spectra and denoted, respectively, by  $\Sigma$ ,  $V$ ,  $H$  and  $X$ . The corresponding spectral components will be denoted by  $s^{(1)}_{\Sigma,i,j}$ ,  $s^{(1)}_{V,i,j}$ ,  $s^{(1)}_{H,i,j}$ ,  $s^{(1)}_{X,i,j}$ . The principles of calculation of the elements (spectral components) of the sub-spectra will be described below.

The 2-nd level morphological spectrum  $\mathbf{S}^{(2)}$  consists of 16 sub-spectra of  $2^{m-2} \times 2^{m-2}$  size denoted by the following pairs of symbols:  $\Sigma\Sigma$ ,  $\Sigma V$ ,  $\Sigma H$ ,  $\Sigma X$ ,  $V\Sigma$ ,  $VV$ ,  $VH$ ,  $VX$ ,  $H\Sigma$ ,  $HV$ ,  $HH$ ,  $HX$ ,  $X\Sigma$ ,  $XV$ ,  $XH$ , and  $XX$ . The sub-spectra will be denoted, respectively, by  $\mathbf{S}^{(2)}_{\Sigma\Sigma}$ ,  $\mathbf{S}^{(2)}_{\Sigma V}$ ,  $\mathbf{S}^{(2)}_{\Sigma H}$ , etc., while the corresponding spectral components will be denoted by  $s^{(2)}_{M,i,j}$  where  $M$  stands for the string of symbols denoting the spectral component under consideration. The sub-spectra of each next-level morphological spectrum are denoted by adding one of symbols  $\Sigma$ ,  $V$ ,  $H$  or  $X$  to the left side of  $M$  denoting a former-level sub-spectrum. The number of sub-spectra constituting the next-level morphological spectrum is thus 4 times higher while the size of the corresponding arrays becomes  $2 \times 2$  times lower. The highest-level morphological spectrum  $\mathbf{S}^{(m)}$  consists of  $2^m \times 2^m$  sub-spectra of  $1 \times 1$  size, i.e. of  $2^{2m}$  real numbers. Consequently, at each spectrum-level a total number of spectral components remains the same.

Let us take into consideration a typical,  $n$ -th level sub-spectrum  $\mathbf{S}^{(n)}_M$ ,  $0 < n \leq m$ . The corresponding array will be divided into  $2 \times 2$  blocks. Let us take into consideration a typical block of this type:

$$B_{p,q}^{(n)} = \begin{bmatrix} s_{p,q}^{(n)} & s_{p,q+1}^{(n)} \\ s_{p+1,q}^{(n)} & s_{p+1,q+1}^{(n)} \end{bmatrix} \quad (2)$$

There will be defined the following operations:

$$\begin{aligned} \Sigma[B_{p,q}^{(n)}] &= s_{p,q}^{(n)} + s_{p,q+1}^{(n)} + s_{p+1,q}^{(n)} + s_{p+1,q+1}^{(n)} & (a) \\ V[B_{p,q}^{(n)}] &= -s_{p,q}^{(n)} + s_{p,q+1}^{(n)} - s_{p+1,q}^{(n)} + s_{p+1,q+1}^{(n)} & (b) \\ H[B_{p,q}^{(n)}] &= -s_{p,q}^{(n)} - s_{p,q+1}^{(n)} + s_{p+1,q}^{(n)} + s_{p+1,q+1}^{(n)} & (c) \\ X[B_{p,q}^{(n)}] &= -s_{p,q}^{(n)} + s_{p,q+1}^{(n)} + s_{p+1,q}^{(n)} - s_{p+1,q+1}^{(n)} & (d) \end{aligned} \quad (3)$$

Finally, for all  $p, q$  one collects:

- \* all elements  $\Sigma[B_{p,q}]$  into one array  $\mathbf{S}_{\Sigma M}^{(m+1)}$ ,
- \* all elements  $V[B_{p,q}]$  into one array  $\mathbf{S}_{VM}^{(m+1)}$ ,
- \* all elements  $\Sigma[B_{p,q}]$  into one array  $\mathbf{S}_{\Sigma M}^{(m+1)}$ ,
- \* all elements  $\Sigma[B_{p,q}]$  into one array  $\mathbf{S}_{\Sigma M}^{(m+1)}$ ,

the size of the arrays being  $2^{n-1} \times 2^{n-1}$ . The last completes the operation of passing from  $\mathbf{S}^{(m)}$  to  $\mathbf{S}^{(m+1)}$ . The symbol  $\mathbf{M}$  thus can be also interpreted as a symbol of operator transforming original image  $\mathbf{u}$  into its morphological spectrum of a given level. It follows from the above-given description of calculation of spectral components that passing from any former-level to the next-level morphological spectrum is a reversible operation on any level the calculation of spectral components consists in summing pixel values with weights  $+1$  or  $-1$  only within a fixed square window. Therefore, it can be simplified due to preliminarily prepared square arrays (called masks) consisting of the signs  $+$  or  $-$ , which superposed on the corresponding square windows indicate, which elements should be summed with the weight  $+1$  or  $-1$ . Typical masks for calculation of the components of  $\mathbf{S}^{(1)}$ , according to the general formulae (3), are shown below:

$$T_{\Sigma}^{(n)} = \begin{bmatrix} + & + \\ + & + \end{bmatrix}, T_V^{(1)} = \begin{bmatrix} - & + \\ - & + \end{bmatrix}, T_H^{(1)} = \begin{bmatrix} - & - \\ + & + \end{bmatrix}, T_X^{(1)} = \begin{bmatrix} - & + \\ + & - \end{bmatrix}.$$

Some examples of the masks for higher-level morphological sub-spectra are given below.

### 3 Invariance of Morphological Spectra with Respect to Image Rotations and Shifts

Examination of morphological spectra leads to several interesting observations, important from image processing purposes point of view.

#### 3.1 Scaling

It can be easily observed that the operator  $\Sigma$  preceding any operator  $\mathbf{M}$  defines an operator  $\mathbf{M}\Sigma$  which results as performing the operation  $\mathbf{M}$  on sums of  $2 \times 2$  couples of pixels. This can be illustrated by a comparison of the masks:

$$T_{XX}^{(2)} = \begin{bmatrix} + & - & - & + \\ - & + & + & - \\ - & + & + & - \\ + & - & - & + \end{bmatrix} \quad T_{XX\Sigma}^{(3)} = \begin{bmatrix} + & + & - & - & - & + & + \\ + & + & - & - & - & + & + \\ - & - & + & + & + & + & - & - \\ - & - & + & + & + & + & - & - \\ - & - & + & + & + & + & - & - \\ - & - & + & + & + & + & - & - \\ + & + & - & - & - & + & + \\ + & + & - & - & - & + & + \end{bmatrix}$$

where  $T^{(2)}_{XX}$  corresponds to a double-cross summing of single pixel values while  $T^{(3)}_{XX\Sigma}$  corresponds to a similar operation performed on quadruples of pixel values.

### 3.2 Symmetries

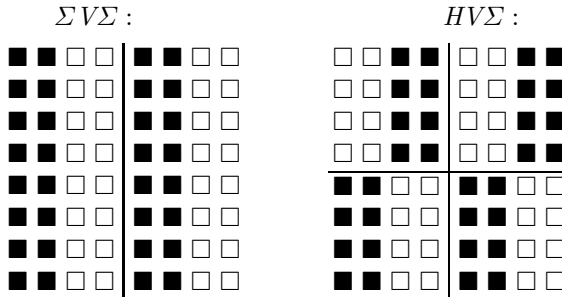
Below, for the sake of simplicity the masks will be denoted directly by the strings of symbols of the corresponding operations, for example, instead of  $T^{(2)}_{XX}$  we shall write directly  $XX$ , instead of  $T^{(3)}_{XX\Sigma}$  the symbol  $XX\Sigma$  will be used, etc.

Examination of masks leads to the following conclusions:

1.  $\Sigma$  operators in the strings  $M$  have no influence on the symmetry or asymmetry of the mask with respect their vertical or horizontal axes;
2. any odd number of  $V$  operators causes asymmetry of the mask with respect to the vertical axis;
3. any odd number of  $H$  operators causes asymmetry of the mask with respect to the horizontal axis;
4. any odd number of  $X$  operators causes asymmetries with respect both to the vertical and horizontal axes;
5. the asymmetries of the above-described types are superposed if several causes act simultaneously.

The above-described rules can be illustrated by the following examples:

$\Sigma V\Sigma$ : asymmetry with respect to vertical axis caused by a single operator  $V$ , and  $HV\Sigma$ : double asymmetry with respect to both axes caused by a simultaneous action of operators  $H$  and  $V$ :

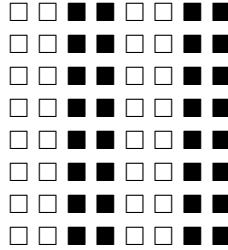


### 3.3 Parallel Shifts

The effect of parallel shifting of the masks by a quarter or a half of the size of window corresponding to a given spectrum level can be reached by using negative masks or by a selection of pairs of the corresponding spectrum components.

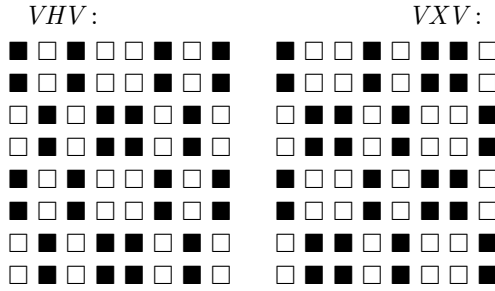
This can be illustrated by the following examples:

The negative mask  $-\Sigma V\Sigma$  has the form:



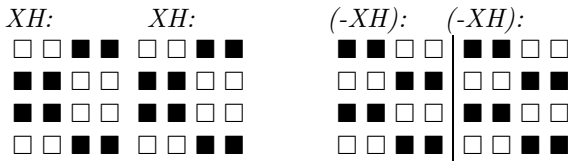
which in fact is similar to this of  $\Sigma V\Sigma$  shifted horizontally by a quarter of the window's size (two pixels).

The masks:



represent, in fact, the same morphological structure shifted horizontally by two pixels.

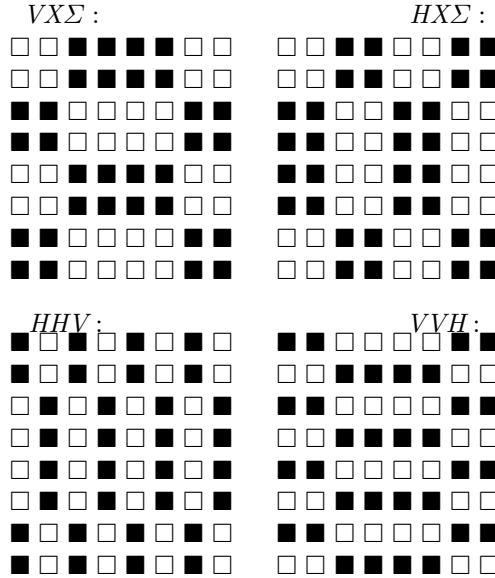
In the case of a mask asymmetrical with respect to its vertical (horizontal) axis taking its negation leads to an effect of parallel horizontal (vertical) shift by a half of the window-size. This becomes visible if two or more adjacent masks of the same type are considered, as shown below:



Due to the above-described properties of morphological spectra it is not necessary in texture analysis to shift physically a specimen in order to make texture recognition independent of the specimen's position in observation area.

### 3.4 Rotations

Substitution of each  $V$  operator by  $H$  operator and vice versa leads to the effect of rotation of the mask by  $-90^\circ$ , as it is shown below:



etc.

### 4 Using to Texture Analysis

When using morphological spectra in analysis of textures it is necessary to specify the requirements for types of textures discrimination. The following problems should be taken into account:

1. What is the highest granularity of textures that should be neglected in the analysis?
2. Is the mean luminance of the texture important or it is to be neglected?
3. Is the texture’s anisotropy to be taken into account or to be neglected?
4. In the case of anisotropic textures analysis - is the texture’s spatial orientation to be taken into account or to be neglected?

According to the way of answering the above-formulated questions the way of using morphological spectra should be chosen.

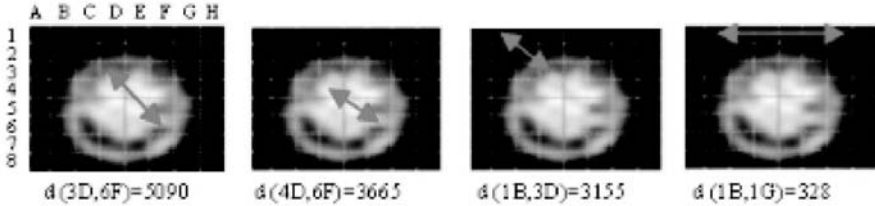
The main criterion for evaluation of a difference between morphological spectra representing two textures is an Euclidean distance between the corresponding vectors of spectral components. However, the distances in different ways should be calculated in the case when collections of components are taken into account or if they are not. Let us assume that only four spectral components are taken into account; they will be denoted by  $[x, y, u, v]$  for one texture and  $[x', y', u', v']$  for the second one. In the case when all spectral components are considered independently the distance will be given by the formula:

$$d(S, S') = \sqrt{(x - x')^2 + (y - y')^2 + (u - u')^2 + (v - v')^2} \tag{4}$$

while in the case when the components are grouped into collections  $(x, y)$ ,  $(u, v)$ , and, respectively,  $(x', y')$ ,  $(u', v')$  the distance will be given by the formula:

$$d(S, S') = \sqrt{[(x + y) - (x' + y')]^2 + [(u + v) - (u' + v')]^2} \quad (5)$$

The difference between the above-mentioned two formulae is substantial.



**Fig. 1.** Distances between spectral vectors of selected window

In order to prove the usefulness of morphological spectra to texture analysis it was taken into consideration a series of NMR images (provided by the Institute of Nuclear Medicine of the Medical Academy in Warsaw ) of human brains. Each image was partitioned into 64 testing windows and for selected pairs of windows full 3-rd order spectra have been calculated. Then Euclidean distances between morphological spectral vectors were calculated. In Fig.1 there are shown examples of images with selected pairs of windows and the distances between the corresponding pairs of spectral vectors. The experiments have shown that the proposed method, in general, can be used to detection and evaluation of differences between the textures in various segments of the brain (say, between their left and right half-sphere).

## 5 Conclusions

Morphological spectra are a new tool for formal description of textures. They have several interesting properties which make them useful in practical applications, like ability to describe multi-level morphological structures, easiness of spectral components calculation, etc. In this paper some other properties have been presented: symmetries of spectral components, ability to describe parallel shifts and rotations of analyzed specimens. It is suggested that morphological spectra should be considered as a complementary tool in computer-aided examination of medical or technical images.

## References

1. Zhu Y.M., Gao Y., Goutte R., Amiel M. (1992). "Textural Boundary Detection Using Local Spatial Frequency Analysis". Proc. 11<sup>th</sup> IAPR International Conference on Pattern Recognition, Hague, vol. III. IEEE Computer Society Press, Los Alamitos; 53-56.
2. J.L Kulikowski., D. Wierzbicka „A Method of Microvascular Systems Analysis Based on Statistical Texture Parameters Evaluation". Biocybernetics and Biomedical Eng., vol. 23. No 3, 2003, pp. 21-37.



3. J.L. Kulikowski, M. Przytulska, D. Wierzbicka. „Recognition of Textures Based on Analysis of Multilevel Morphological Spectra”. IFMBE Proceedings, Vol. 14. World Congress on Medical Physics and Biomedical Engineering, Seoul, 2006, pp. 2164-2167.
4. J.F Haddon., J.F. Boyce (1992). “Texture Segmentation and Region Classification by Orthogonal Decomposition of Cooccurrence Matrices”. Proc. 11<sup>th</sup> IAPR International Conference on Pattern Recognition, Hague, vol. I. IEEE Computer Society Press, Los Alamitos: 692-695.
5. T. Ojala, M. Pietikajnen. „Unsupervised Texture Segmentation Using Feature Distributions, Texture Analysis Using Pairwise Interaction Maps”, *Image Analysis and Processing*, 9<sup>th</sup> International Conference, ICIAP'97, Florence, Proc. vol. I. (A. Del Bimbo ed.), 1997, pp. 311-318.
6. G. Loum, J. Lemoine et al. “An Application of Wavelet Transform to Texture Analysis”. Proc. of the 9<sup>th</sup> Scandinavian Conference on Image Analysis, vol. 1, Uppsala, 1995, pp. 583-590.
7. Y. Xiaohan, J. Yla-Jaaski. “Unsupervised Texture Segmentation Based On the Modified Markov Random Field Model”. Proc. 11<sup>th</sup> IAPR International Conference on Pattern Recognition, Hague, vol. III. IEEE Computer Society Press, Los Alamitos, 1992, pp.88-91.
8. T.G Smith., G.D. Lange. „Biological Cellular Morphometry-Fractal Dimensions, Lacunarity and Multifractals”. *Fractals in Biology and Medicine*, vol. II (Losa G.A., Merlini D. et al. Eds.). Birkhauser, Basel, 1998, pp. 30-49.

---

# Breast Contour Detection for the Aesthetic Evaluation of Breast Cancer Conservative Treatment

Jaime S. Cardoso<sup>1</sup> and Maria J. Cardoso<sup>2</sup>

<sup>1</sup> INESC Porto, Faculdade de Engenharia, Universidade do Porto, Portugal  
jaime.cardoso@inescporto.pt

<sup>2</sup> Faculdade de Medicina, Universidade do Porto, Portugal  
mjcard@med.up.pt

**Summary.** Cosmetic assessment of breast cancer conservative treatment (BCCT) plays a major role in the evaluation of this form of treatment. Objective assessment methods are being preferred to overcome the drawbacks of subjective evaluation. A recent computer-aided medical system was developed to objectively and automatically perform the aesthetic evaluation of the result of BCCT. In order to extract relevant features from the image, the detection of the breast contour is necessary. In this paper an algorithm based on the shortest path on a graph is proposed to detect the breast contour. The proposed method was applied to 300 breast images with an accuracy of 98%.

## 1 Introduction

Breast cancer conservative treatment (BCCT) has been increasingly used over the last few years as a consequence of its more acceptable cosmetic outcome when compared with mastectomy, but with identical oncological results. Although considerable research has been put into BCCT techniques, diverse aesthetic results are common, highlighting the importance of this evaluation in institutions performing breast cancer treatment, so as to improve working practices.

Traditionally, aesthetic evaluation has been performed subjectively by one or more observers [1]. The categorization of the aesthetical result relies on the complex interplay of various factors, subjectively estimated and combined by observers through visual inspection. Taking into account the inherent subjectivity of any human decision, the final evaluation of aesthetic result performed by observers is questionable. In fact, this form of assessment has been shown to be poorly reproducible [2], which creates uncertainty when comparing results between studies. It has also been demonstrated that observers with different backgrounds evaluate cases in different ways [3].

Objective methods of evaluation have emerged as a way to overcome the poor reproducibility of subjective assessment. Initially they consisted only of measurements between identifiable points on patient photographs [4, 2]. The correlation of objective measurements with subjective overall evaluation has been

reported by several authors [2]. Latter, the overall cosmetic outcome was simply the sum of the individual scores of subjective and objective individual indices [5]. Recently, a computer-aided medical system was developed to objectively and automatically perform the aesthetic evaluation of the result of the intervention [6]. The development of this system entailed the automatic extraction of several features from the photographs (Fig. 1), capturing the main factors with impact on the overall cosmetic result: breast asymmetry, skin colour change due to the treatment and surgical scar visibility. In a second phase, a support vector machine classifier was trained to predict the overall cosmetic result from the recorded features [6].



(a) Facing, arms down



(b) Facing, arms up

**Fig. 1.** Positions used in the photographs

In order to extract the identified relevant features from the image, the detection of the breast contour is necessary. In this work, an algorithm for the automatic detection of the breast contour is described. The proposed approach is formulated as the solution to the shortest path between the end points of the breast contour, after conveniently modelling the image as a weighted graph. The algorithm has been implemented in a semi-automatic software, where the user just has to identify some key points on the image: breast contour end points and nipples position. The software automatically finds the contours, extracts relevant features and outputs a predicted overall cosmetic assessment (*excellent, good, fair, poor*).

## Previous Work

Although several studies have addressed breast contour detection, they all concern the detection on digital mammograms, on a side view of the breast — see [7] and references within. Our preliminary research was based on the polynomial modelling of the breast contour. After a pre-processing step to extract image edges (using e.g. a canny detector), a 2D polynomial  $P_n(x, y)$  was then fitted by a regression method to the contour pixels. Suitable degrees of the polynomial were evaluated. Nevertheless, results were unsatisfactory as the regression was biased by the high number of contour pixels not belonging to the breast and/or the polynomial had not enough flexibility to adjust to the breast contour. Note

that the breast contour may be severely deformed, departing from the typical round or tear drop shape.

A second line of experiments was conducted with active contours. Active contours have also been successfully applied to breast contour detection on mammograms [8]. However, results were still behind the expected. Moreover, the computation time was excessive. The minimization of the user involvement and the total user's time required for contour detection is obviously paramount.

## 2 A Shortest Path Approach to Contour Detection

Knowing the two endpoints of the breast contour (inputted by the user), we are left with the problem of finding the path between both endpoints that goes through the breast contour. As the interior of the breast itself is essentially free of edges, the path we are looking for is the shortest path between the two endpoints, **if** paths (almost) entirely through edge pixels are favoured.

### 2.1 Definitions and Notation

A *graph*  $G = (V, A)$  is composed of two sets  $V$  and  $A$ .  $V$  is the set of nodes, and  $A$  the set of arcs  $(p, q)$ ,  $p, q \in V$ . The graph is *weighted* if a weight  $w(p, q)$  is associated to each arc, and it is called a *digraph* if the arcs are directed, i.e.,  $(p, q) \neq (q, p)$ . A path from  $p_1$  to  $p_n$  is a list of unique nodes  $p_1, p_2, \dots, p_n$ ,  $(p_i, p_{i+1}) \in A$ . The *path cost* is the sum of each arc weight in the path.

In graph theory, the shortest-path problem seeks the shortest path connecting two nodes; efficient algorithms are available to solve this problem, such as the well-known Dijkstra algorithm [9].

### 2.2 Proposed Algorithm

To detect the breast contour we propose a two step approach:

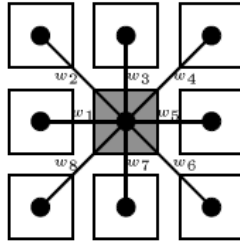
1. Apply an edge detector to the original image. The resulting binary image enhances the points of interest.
2. Detect the breast contour on the edge image, by finding the shortest path between the two endpoints defined by the user.

We now detail this second step.

Starting by modelling the edge image as a graph, match a node to each pixel. Connect two nodes with an arc on the graph iff the corresponding pixels are neighbours (8-connected neighbourhoods) on the image. The weight of each arc is a function of pixels values and pixels relative positions — see Fig. 2:

$$w_i = \begin{cases} f(p, q_i) & \text{if } q_i \in \text{4-connected neighbourhood of } p \\ h(p, q_i) & \text{if } q_i \notin \text{4-connected neighbourhood of } p \end{cases}$$

In this work we set  $h(., .) = \sqrt{2}f(., .)$ .



**Fig. 2.** Arc weight between two pixels

Now, because we want to favour paths through edge pixels, we set

$$f(p, q) = \begin{cases} c_1 & \text{if both } p \text{ and } q \text{ are edge pixels} \\ c_2 & \text{otherwise} \end{cases}$$

with  $c_2 > c_1$ . In this work  $c_1$  and  $c_2$  were experimentally determined as 2 and 32, respectively. Note that  $c_1$  must be set greater than zero in order to also favour the smallest path, when more than one exists through edge pixels only. Finally, the solution to the shortest path problem will yield the intended breast contour.

### 2.3 Algorithm Generalization

The proposed paradigm can be conveniently generalized. The application of an edge detector in the first step can miss to detect segments of the breast contour. This is especially true for women with small breasts (leading to weak contours) or when the breast is severely deformed with the excision of a large sample of tissue. A natural improvement is to replace the binary image outputted by the edge detector with a richer gradient image. Now, the shortest path algorithm should try to follow pixels with high gradient values. The  $f(\cdot, \cdot)$  and  $h(\cdot, \cdot)$  functions have to be properly generalized. A simple strategy is to set  $f(p, q) = \hat{f}(255 - \min(p, q))$ , where  $\hat{f}(\cdot)$  is a monotonically increasing function. Note that this more general setting has the binary case as a particular instantiation. To summarize, the proposed general framework to find the contour between two endpoints encompasses:

- A gradient computation of the original image. In a broader view, this can be replaced by any feature extraction process that emphasizes the pixels we are seeking for.
- Model the gradient image with a weighted graph, assigning to the weight between two neighbour pixels the cost  $w(p, q) = f(p, q)$  or  $w(p, q) = h(p, q)$ , as described before.

## 3 Results

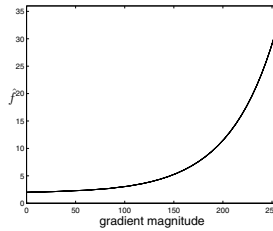
The proposed methodology was applied to a set of images from 150 patients. For each patient two positions were assessed: facing, arms down and facing,

arms up (see Fig. 1). Before applying the breast contour detection, each image was downsized to a constant width of 768, keeping the aspect ratio. Only the luminance information was used in the analysis.

The binary model makes use of the Canny edge detector [10] on the first step, as implemented in the Open CV library [11]. The low and high thresholds were experimentally tuned to 32 and 128, respectively. In the graph derived from the binary image, costs were assigned as previously stated: 2 when both incident pixels are edge pixels, 32 otherwise. The gradient model is based on the Sobel operator. The Sobel operator is applied on the  $x$  and  $y$  directions; from the computed values,  $S_x$  and  $S_y$ , the magnitude of the gradient is estimated as  $z = \sqrt{S_x^2 + S_y^2}$ . Costs were assigned based on an exponential law:

$$\hat{f}(z) = \alpha \exp(\beta z) + \delta, \quad \alpha, \beta, \delta \in \Re$$

The parameters  $\alpha, \beta, \delta$  were chosen to yield  $\hat{f}(0) = 2$  and  $\hat{f}(255) = 32$ , leading to the same range of costs as the binary model. The third degree of freedom was experimentally tuned. The adopted transformation was (see also Fig. 3)  $\hat{f}(z) = 0.15 \exp(0.0208z) + 1.85$ .



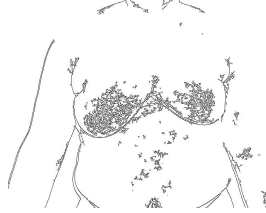
**Fig. 3.** Transformation  $\hat{f}(\cdot)$

Fig. 4 shows three breast contour detection results obtained with the binary model. The end points specified by the user are shown as small circles. The method successfully completed the gaps between detected edge pixels and ignored spurious edge pixels.

Problems with the binary model are noticeable when the edge detector misses to detect significant portions of the breast contour. The shortest path algorithm is misled to follow non-breast pixels — see Fig. 4(f) and Fig. 4(i). The gradient model presents a more robust behaviour, correctly following the breast contour, even in the presence of contours not well defined — see Fig. 5. While in the right breast of Fig. 4(f) the binary model followed a straight path to fill the gap corresponding to a soft edge, the gradient model correctly followed the contour. Similarly, in Fig. 4(i) the binary model was completely deceived by the presence of edges around the nipple and their absence along the breast contour; the gradient model properly identified the contour.



(a) Patient #31



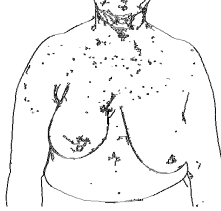
(b) Canny edge detector



(c) Model output



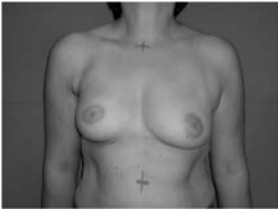
(d) Patient #11



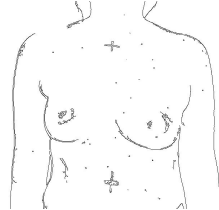
(e) Canny edge detector



(f) Model output



(g) Patient #27



(h) Canny edge detector

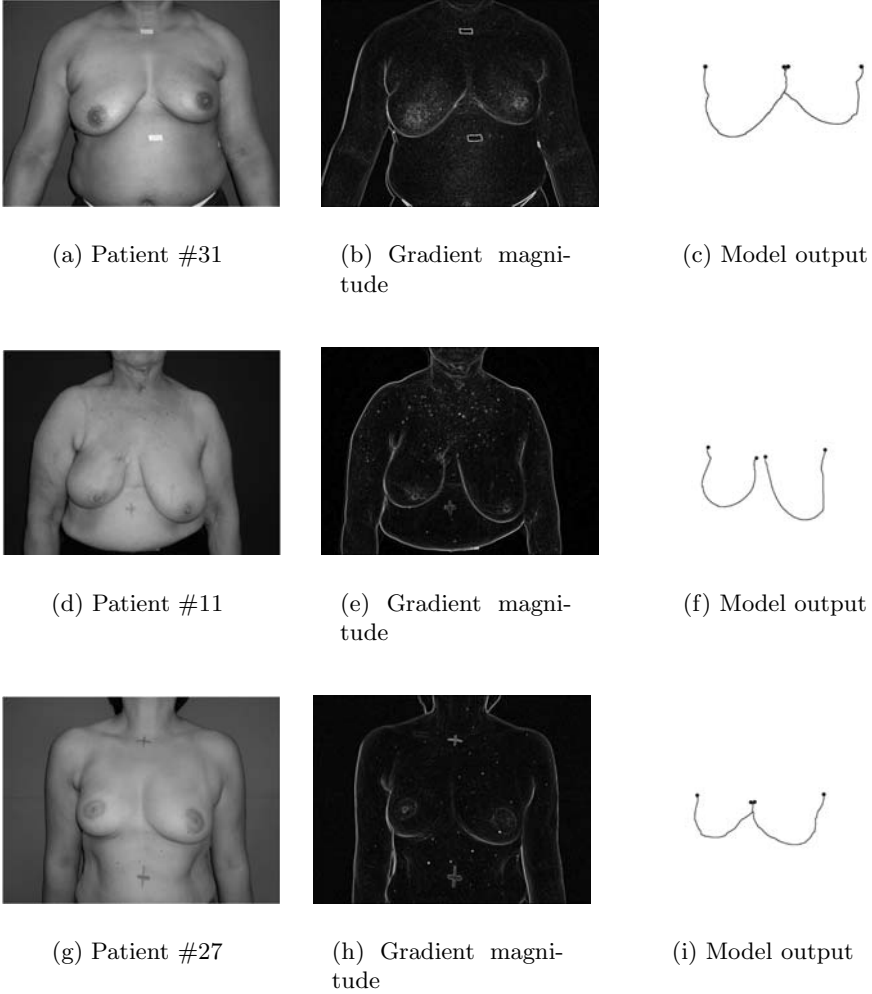


(i) Model output

**Fig. 4.** Results for the binary model

Finally, the evaluation of the breast contour detected in the photographs was performed by an observer, in the photograph dataset. The accuracy of contour detection is presented in Table 1, discriminated by arms down/ arms up and treated/untreated breast.

Results bring to light the superiority of the gradient model over the binary model. Moreover, both models behave worse on the treated breasts than on the untreated breasts, most likely due the deformations resulting from the treatment. It is also visible a better performance in the arms down position, this time due to the better definition of the breast contours.



**Fig. 5.** Results for the gradient model

**Table 1.** Performance of the proposed method

	arms down		arms up		average performance
	treated	untreated	treated	untreated	
binary	0.94	0.97	0.91	0.94	0.94
gradient	0.98	1.00	0.96	0.98	0.98

## 4 Conclusions

We have described how the breast contour can be found as the solution to the shortest-path problem in the graph theory framework, after conveniently



modelling the image as a weighted graph. The proposed approach is computationally efficient, taking only a few milliseconds to detect a breast contour, in a standard PC. This is an important aspect in user interactive applications. Preliminary results also indicate an excellent performance. The proposed framework can potentially be applied on similar edge tracking problems.

## References

1. J. R. Harris, M. B. Levene, G. Svensson, and S. Hellman, "Analysis of cosmetic results following primary radiation therapy for stages i and ii carcinoma of the breast," *International Journal of Radiation Oncology Biology Physics*, vol. 5, pp. 257–261, 1979.
2. D. R. H. Christie, M-Y. O'Brien, J. A. Christie, T. Kron, S. A. Ferguson, C. S. Hamilton, and J. W. Denham, "A comparison of methods of cosmetic assessment in breast conservation treatment," *Breast*, vol. 5, pp. 358–367, 1996.
3. M. J. Cardoso, A. C. Santos, Jaime S. Cardoso, H. Barros, and M. C. Oliveira, "Choosing observers for evaluation of aesthetic results in breast cancer conservative treatment," *International Journal of Radiation Oncology, Biology and Physics*, vol. 61, pp. 879–881, 2005.
4. E. V. Limbergen, E. V. Schueren, and K. V. Tongelen, "Cosmetic evaluation of breast conserving treatment for mammary cancer. 1. proposal of a quantitative scoring system," *Radiotherapy and oncology*, vol. 16, pp. 159–167, 1989.
5. S. K. Al-Ghazal, R. W. Blamey, J. Stewart, and A. L. Morgan, "The cosmetic outcome in early breast cancer treated with breast conservation," *European journal of surgical oncology*, vol. 25, pp. 566–570, 1999.
6. Jaime S. Cardoso and Maria J. Cardoso, "Towards an intelligent medical system for the aesthetic evaluation of breast cancer conservative treatment," *Artificial Intelligence in Medicine*, vol. 40, pp. 115–126, 2007.
7. H. Mirzaalian, M. R. Ahmadzadeh, and F. Kolahdoozan, "Breast contour detection on digital mammogram," in *Proceedings of IEEE International Conference on Information and Communication Technologies (ICTTA'06)*, 2006, vol. 1, pp. 1804–1808.
8. Michael A. Wirth and Alexei Stapinski, "Segmentation of the breast region in mammograms using snakes," in *Proceedings of the First Canadian Conference on Computer and Robot Vision (CRV04)*, 2004, pp. 385–392.
9. E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
10. J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.
11. "Intel corporation, open source computer vision library," <http://www.intel.com/technology/computing/opencv/index.htm>.

---

# Simple EEG Driven Mouse Cursor Movement

Jan Kněžík and Martin Dražanský

Brno University of Technology, Faculty of Information Technology,  
Bozotechnova 2, CZ-61266, Brno  
knezik@ro.vutbr.cz

**Summary.** This paper dwells on suggestion of direction control of computer mouse by electroencephalographic (EEG) device incorporated, which makes it possible for the user to affect the direction of the cursor's movement on the screen by the frequency of brain's oscillation. The motivation for solution of this problem is the effort to help to handicapped people to communicate with surrounding world. Described approach uses technique called operant conditioning [1] and in the simplest version is based on comparison of signal magnitude in two narrow neighbouring bands. Computed difference influences direction and speed of cursor's motion. Promising results of one axis control were achieved and described approach will be considered in more complex brain-computer interface.

## 1 Introduction

The issue of computer control using human mind became one of very popular topics among public and scientific community in recent years. Nowadays there exist systems enabling person to walk along the street of city created by virtual reality or helping handicapped people to communicate with surrounding environment by means of computer. However, beside strong progression of Brain-Computer Interfaces (BCI), these systems are faced with many problems to be solved. Up to now, there does not exist absolutely universal, currently available and especially fully accurate system managing all tasks connected with interaction between human and computer.

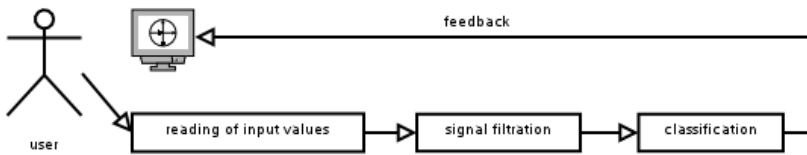
Because the user interfaces of modern computers are heavily based on usage of keyboard and mouse, very interesting thing will be replacing these peripherals with device capable to detect user commands directly from the brain. Emulation of computer mouse is more suitable for this purpose than keyboard one since less information is needed and right detection of desired orders is much simpler. An individual is able to control direction of computer mouse cursor and sometimes click and drag-and-drop options are also included. There are several approaches to achieve this behaviour. Operant conditioning method lets the user learn how to control the movement of mouse cursor by means of tight feedback loop. Cursor moving takes some effort, especially on the beginning of the training. Other methods use pattern recognition in order to recognize signals associated with the mental states of untrained individuals. Usually, in this case the computer is

being taught by the user to recognize his or her appropriate commands with one of the machine learning algorithms. The best approach should be the estimation of commands without any previous learning sessions, but it demands very strong knowledge of processes which take place in human brain.

## 2 Background

Operation of typical BCI interface can be usually separated into four following parts [2], as can be seen on Fig. 1:

- Acquisition of input data – bioelectrical activity of brain is recorded by the EEG device and sent to the computer
- Signal preprocessing – undesirable artifacts are suppressed and signal waveform is modified for simpler extraction of required information
- Classification – own acquisition of information using Neural Networks, Hidden Markov Models, Bayesian Networks and other methods of Artificial Intelligence
- Feedback – presentation of classifier's result by means of computer, therefore this process affects user of described system



**Fig. 1.** Diagram of BCI interface

Data acquisition is the most important part of the whole BCI system. The elementary attribute by which the BCI systems differ one from another is the way they obtain the EEG signal. There are two possible placements of electrodes. Invasive electrodes are placed directly inside head on surface of brain cortex. Signal attenuation and distortion connected with its passing through the skull and skin is completely removed. This approach is suitable if the interface is being used continuously (i.e. for handicapped people) or high level of input signal is needed. The other of two ways to record EEG is noninvasive placement when electrodes are placed on skin with conductive gel or paste. This is the most useful solution for occasional usage when only external monitoring is wanted. In the most cases the data acquisition is done via high-end devices with the costs in the range USD\$20,000-250,000, but this price is considerably expensive for ordinary users. Fortunately, there are several amateur groups of enthusiasts like OpenEEG [3] who are developing low-cost electroencephalograph intended for home usage. This project is not focused only on hardware, but also comes

with set of useful programs and thus it represents very interesting platform for early experiments.

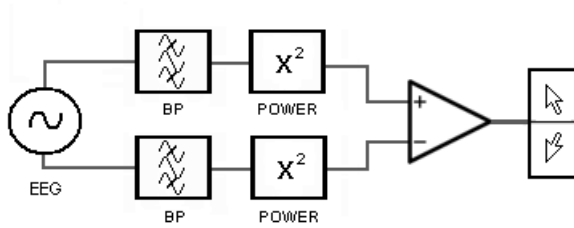
Obtained signal is preprocessed in the next stage, where the main task is removing artifacts and feature generation. Artifacts [4] refer to any disturbance in the recording that does not arise from the brain. It includes muscle activity and eye movement, 50/60 cycle mains hum, electrode movement and pop, vascular (electrocardiographic) artifacts and drip artifacts. All these unwanted signals are usually successfully suppressed by 4-25 Hz bandpass filter. Typical features generated from input signal are values of spectral power of signal in several bands. Fast Fourier Transform (FFT) is often used as classic and proven algorithm for doing this job. In this case one have to separate samples into the blocks called time windows and run FFT algorithm within this windows. Time resolution is given by length of these windows. Better is to use wavelet transform, which has the advantage of good time localisation, because it reads input samples continuously and processes them by a bank of filters instead of making signal windows. Other popular signal features are phase coherence and band power difference between the two channels.

Command classification comes after signal preprocessing. Main problem with pattern recognition is handling input uncertainty and separation of short distance input vectors. Various algorithms have been developed on basis of soft computing (Neural Networks, Fuzzy Clustering), theory of propability (Bayesian Networks, Hidden Markov Models) and linear algebra (Support Vector Machines [5]). Because there is no proof that one mentioned method is better than the others for EEG signal classification, all of them are widely used. Drawback of these algorithms is demand of large amount of input information for perfect decision, hence classified signal is usually obtained from numerous input EEG channels.

### 3 The Proposed Approach

Described design of mouse control system comes from the concept totally different from the most other solutions of the mouse control problem. Important inspiration for me was piece of knowledge from the field of biofeedback and psychostimulators that an individual is able to voluntary affect level of bioelectric activity of his or her brain in a specific frequency band on condition that information about the speed and character of measured frequency change is provided. Therefore no generation of some accurate signal pattern for cursor moving is needed and frequency swing is the only guide for this system.

The basic principle (see Fig. 2) is relatively simple and consists in continous comparison of signal magnitude (power in pass-band) value of the two neighbouring bands obtained by bandpass filtering. The result of this comparison will influence direction of cursor movement in given axis of coordinate system (e.g. upward or downward). Frequency range of both bands is intently set very close and filters let through only a narrow range of frequencies in order to allow user



**Fig. 2.** Block diagram of cursor actuating

to control cursor's movement easily by means of "retuning" of brain waves. List of brain wave bands and description of their connection to mental states can be seen in Tab. 1.

**Table 1.** Relation between brain wave frequencies and mental states

Band	Frequency	Mental state
Delta	1-3 Hz	deep sleep
Theta	4-7 Hz	light sleep, daydreaming, meditation
Alpha	8-12 Hz	basic consciousness (closed eyes, relaxation)
SMR	12-15 Hz	relaxed attention, autoregulation
Beta	15-20 Hz	concentrated attention
Beta 2	21-30 Hz	tension, irritation, anxiety
Gamma	31 Hz and above	hyperexcitation

For the interface implementation application BrainBay [6] was used which is being developed in connection with the OpenEEG project by group of developers around Christoph Veigl. This application provides a large scale of functions and it is well suited for experiments with bioelectric signal processing. As can be noticed on Fig. 3, a new bandpass filter with frequency range between the two already existing ones was added to the design. Its role is to make easier to stop the cursor's moving hence increased activity in this band is neutral to the movement direction. Samples are sent from the input interface to 3 blocks analysing signal power level in given bands (mean frequency is displayed above block, bandwidth is in all cases set to 0.2 Hz). Power level outputs are connected with the comparators, which in case of fulfillment of displayed condition let pass input value A to the output, in opposite case their output is disabled. Outputs of both comparators are disabled in the moment when the highest power level is in the middle band and the cursor is supposed to stop. Expression evaluator subtracts both values from comparator and polarity of result is used for control cursor's direction. Constant placed in denominator of expression slows down the speed of cursor's motion.

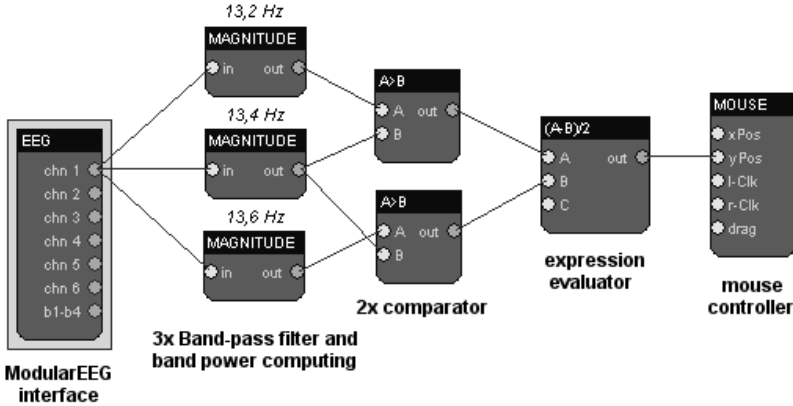


Fig. 3. Design used in BrainBay

### 4 Results and Conclusion

Several experiments with ModularEEG device [3] were performed to test designed circuit. In all tests tin electrodes and conductive paste Ten20 were used. Described design has been found to be very successful for cursor’s motion control for one axis. Two axis control with two independent channels of electroencephalograph has been tried too, but this approach demands high effort from user to control the movement properly. Experiments with various frequency ranges shown that the best way is to work with frequencies from 7 Hz to about 18 Hz. Especially good results have been obtained around 10 Hz and 15 Hz (see Tab. 2). This observation is corresponding with [2], where values from alpha and low beta bands were shown as the most predictive features for many participants of the research. Thanks to narrow bandpass filters at input stage this design has very

Table 2. Description of system behaviour in various brainwave bands

Band	Cursors’s stability	Controllability
Delta	very bad	very bad
Theta	very bad	poor
Alpha	good	good
SMR	poor	poor
Beta	very good	very good
Beta 2	good	poor
Gamma	good	poor

good signal distortion tolerance. On the other hand high dependence on current mental state of an individual and slower reaction time at some occurrences can

be considered as drawbacks of presented system. This problem could be solved by initial spectral analysis and automatic setup of parameters. Described approach gives promising results and will be used to build more complex BCI interface in further work.

## References

1. Operant conditioning. Wikipedia.  
[http://en.wikipedia.org/wiki/Operant\\_conditioning](http://en.wikipedia.org/wiki/Operant_conditioning)
2. Lee J, Tan D (2006) Using a Low-Cost Electroencephalograph for Task Classification in HCI Research. Proceedings of the 19th annual ACM symposium on User interface software and technology, 81–90
3. OpenEEG Project. <http://openeeg.sourceforge.org>
4. Hughes J (1994) EEG in Clinical Practice. Butterworth-Heinemann, Boston London Oxford Singapore Sydney Toronto Wellington
5. Yong Y, Hurley N, Silvestre G (2005) Single-Trial EEG Classification for Brain-Computer Interface using Wavelet Decomposition. Proceedings of EUSIPCO 2005, 81–90
6. BrainBay Application. <http://www.shifz.org/brainbay/>

---

# The Prediction of Fetal Outcome by Applying Neural Network for Evaluation of CTG Records

Michal Jezewski<sup>1</sup>, Janusz Wrobel<sup>2</sup>, Krzysztof Horoba<sup>2</sup>, Adam Gacek<sup>2</sup>,  
Norbert Henzel<sup>1</sup>, and Jacek Leski<sup>1</sup>

<sup>1</sup> Institute of Electronics, Silesian University of Technology, 16 Akademicka Str.,  
44-100 Gliwice, Poland  
mjezewski@polsl.pl

<sup>2</sup> Biomedical Informatics Department, Institute of Medical Technology and  
Equipment, 118 Roosevelt Str., 41-800 Zabrze, Poland  
jezewski@itam.zabrze.pl

**Summary.** Cardiotocography (CTG) as a simultaneous recording of fetal heart rate, uterine contractions and fetal movement activity is a primary method for the assessment of fetal condition. At present, computerized fetal monitoring systems for on-line automated signal analysis are widely used. But effective techniques enabling conclusion generation are still being searched, and neural networks (NN) seem to be particularly attractive. In the presented work a number of investigations were carried out concerning application of NN when quantitative parameters describing fetal CTG signal - input variables - were used for prediction of fetal outcome (normal or abnormal). We tested how the efficiency of NN classification could be influenced by different modification of inputs, by interpretation of fetal outcome definition (output) as well as by various modifications of learning data sets. The obtained results gave a good background for application of the proposed classification tool within computer-aided fetal surveillance systems.

## 1 Introduction

Cardiotocographic monitoring (CTG) is a primary biophysical method for assessment of a fetal state. It consists in simultaneous registration of fetal heart rate (FHR), uterine contraction activity and fetal movement signals. Classical visual analysis of traces from bedside monitor is rather difficult. The interpretation is subjective and is characterized by high interobserver and intraobserver disagreement. The repeatable and objective assessment of the fetal state is of particular importance for high risk pregnancy, when thanks to the early diagnosis, appropriate management can be carried out.

The fetal heart rate variability contains the most important diagnostic information, which is hidden for a naked eye, but can be quantitatively described with a help of dedicated computer-aided systems. Therefore, nowadays computerized fetal monitoring systems have become a standard in clinical centres [2, 4]. The systems archive and dynamically present the incoming data, and additionally



they offer the on-line automated quantitative signal analysis. But effective methods, which could enable conclusion generation, are still being searched. The promising approach seems to be the application of artificial neural networks (NN). Properties of handling complex data sets, capability of learning and generalization, and a distributed pattern recognition process, make neural networks particularly attractive for medical application – also for the fetal outcome assessment [1, 3, 8, 9]. In the proposed work we tried to answer some important questions, which occurred during application of NN, when the quantitative parameters describing fetal cardiocograms – input variables – were used for prediction of the fetal outcome as being normal or abnormal. The aim was focused on a testing how the quality of neural network classification could be affected by different configurations of the input variables, by interpretation of the definition of the fetal outcome being the output as well as by various learning data sets. These aspects are very characteristic for this particular approach connected with fetal surveillance systems for obstetrics departments [5, 6, 7].

## 2 Methodology

Using neural network to predict the fetal outcome meets some problems that were considered in the proposed methodology (Fig. 1). Fetal outcome can be evaluated just after the delivery basing on logical function (mostly OR) of several attributes describing the newborn baby. Whereas in the course of pregnancy the fetal outcome is predicted basing on the parameters provided by quantitative analysis of CTG signals being acquired during fetal monitoring.

Two representations of the neural network output were compared in relation to the classification efficiency. In the first approach the one NN predicted directly the fetal outcome, while in the second approach, the four NNs were developed to classify the particular outcome attributes as normal or abnormal, and resulted fetal outcome was obtained as the OR function of the four NN outputs. The optimal number of the quantitative CTG parameters was found basing on the analysis of NN sensitivity to the inputs. Finally, we investigated the influence of various data grouping within the learning sets on the NN prediction quality.

The archive of computerized fetal surveillance system [4] included parameters of quantitative description of CTG traces and associated medical data referring to particular patients and their newborns. The research material comprised 1274 traces from 341 patients (Fig. 2). After several steps of data cleaning, 685 traces from 189 patients were qualified as the final database. The set of 21 parameters of quantitative description of the CTG record (usually 30÷60 minutes long) was chosen as input variables for NN. Nineteen parameters were describing the fetal heart rate signal in time domain: baseline, acceleration/deceleration patterns, beat-to-beat variability (six various indices) as well as the long-term variability (eight indices). Additionally, the frequency of recognized uterine contractions and fetal movements were used.

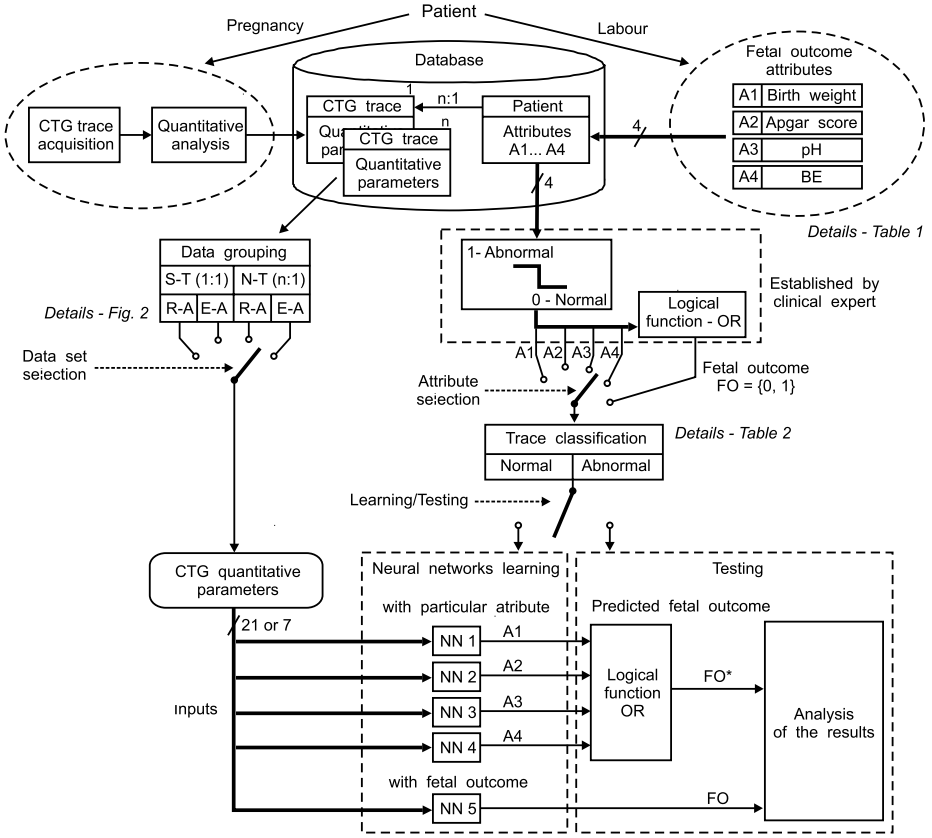


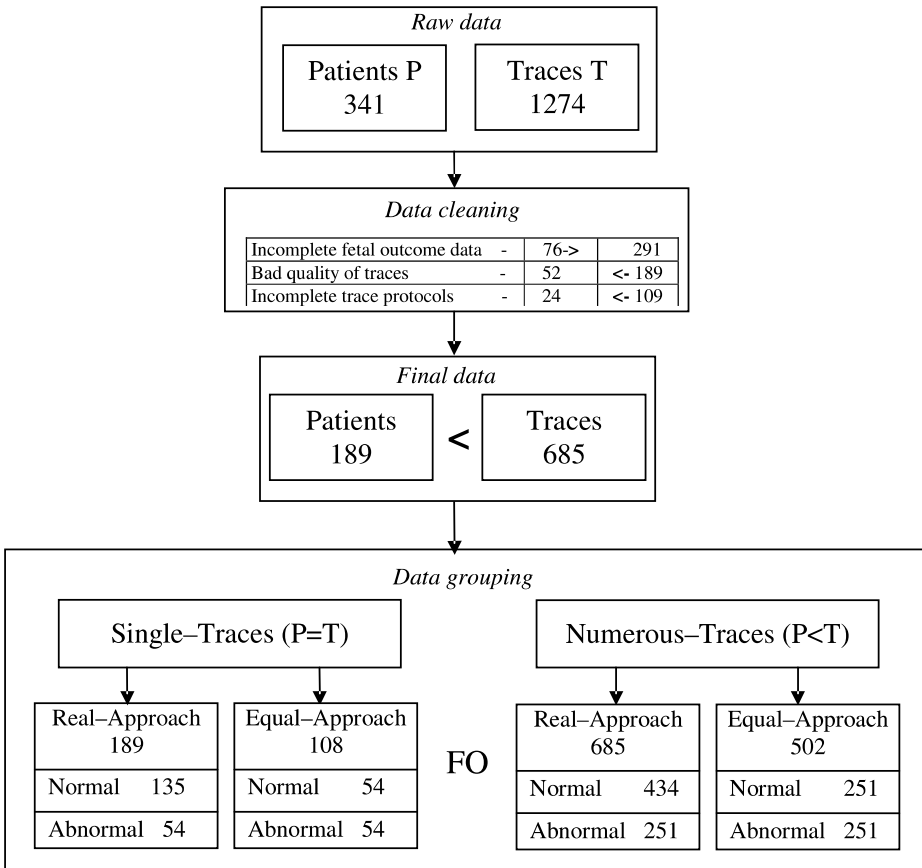
Fig. 1. General diagram of the applied investigation procedure

Table 1. Fetal outcome attributes and criteria for abnormal value

Attribute	Definition $A_i$	Abnormal ( $A_i=1$ ) if:
A1	Neonatal birth weight, just after the delivery	Birth weight < 10 centiles for a given gestational age
A2	Five minute Apgar score (visually assessed newborn state in range 1–10 points)	Apgar score (after first minute) < 7
A3	Umbilical artery pH at birth	pH < 7.2
A4	Umbilical artery base excess (BE) at birth	BE > 12 mEq/l

### 2.1 Outputs Representation

At the beginning, four separated networks (NN1÷NN4) were designed. Each of them had a single two-state output (abnormal or normal) and was dedicated for



**Fig. 2.** Overview of several stages of data cleaning and grouping

prediction of one of the four crucial attributes describing fetal outcome just after the delivery (Tab. 1). In the consecutive step (Fig. 1) the neural network NN5 was learned basing on the so called final fetal outcome, which was defined as two-state value: abnormal or normal. The assessment was made by clinical experts using the data included in the medical forms, describing the patient’s health history. In obstetrics it is assumed, that verification of the fetal state assessment during the course of pregnancy, i.e. when the signals are being recorded, can be made only retrospectively using the real fetal outcome. This is a result of analysis of the newborn state and the history of labour progress evaluated just after the delivery. Such retrospective verification is possible because the fetal state can not rapidly change during the pregnancy.

The results of trace classification according to the fetal outcome (from NN5) were compared with the classification carried out using the outputs of the four separately designed networks. Usually the fetal outcome is defined by clinicians

as abnormal if at least the value of one attribute is outside the physiological range, so the OR function was applied to outputs of networks NN1÷NN4. However, various function can be used, and the most restricted one is the AND function.

## 2.2 Inputs Configuration

A large number of various parameters for quantitative description of CTG trace caused an attempt to reduce them [5]. For the NN5 network, a sensitivity ratio for each of input variables was computed. It represents the ratio of the network's error with the rejected given input variable to the error of network trained with all input variables. The new set of seven selected inputs, with the highest sensitivity ratios within the particular categories of FHR variability, was created. The new NN5 network was designed with only the selected seven input variables.

## 2.3 Data Grouping

One woman could have had several cardiotocographic traces if she was in high-risk pregnancy. That caused there were patients with several CTG traces. More than 50% of all the CTG traces belonged to patients which had from one to five traces recorded. Taking this into account, two main groups of traces called Numerous-Traces and Single-Traces were created (Fig. 2). Numerous-Traces group consists of CTG traces, where each trace, never mind if it belongs to the same patient, is treated as a separate case. Single-Traces group has CTG traces fulfilling the relation that a one trace, registered as closest to the delivery as possible, is assigned to one patient. In this group the number of traces is reduced from 685 to 189. This reduction can prevent from too much influence of individual patient's features, e.g. when CTG traces from a given patient have specific (very high or very low) FHR variability. Comparison of results of networks designed for both sets was done.

Our database included a small number of CTG traces for which the fetal outcome was assigned as abnormal (Tab. 2). Such distribution represents the real situation in high-developed countries, where the fetal outcome is normal in most of the cases. Situations, when sizes of different classes in the learning subset extremely differ, are considered as unfavourable and the equal classes are recommended. On the other hand, class sizes corresponding to the real distributions can give more correct classification by neural network. Therefore data were grouped in two ways (Fig. 2), and the results were compared. The first one was called the Real-Approach, in which the classes were left with original sizes. In another one called Equal-Approach, the classes with exact the same number of traces were used, and this was done by removing some randomly selected traces belonging to the normal class.

The Statistica Neural Networks 7.1 (StatSoft, Inc.) software was used for the development of artificial neural networks. We used the IPS tool for neural network designing. The best architectures of radial basis function type networks were searched using all of the 21 inputs. The limit for searching time was

**Table 2.** Detailed description of results of data grouping as Single- and Numerous-Traces, and for the Real- and Equal-Approach

		Real-Approach					Equal-Approach				
		A1	A2	A3	A4	FO	A1	A2	A3	A4	FO
Single-Traces	Normal	176	149	170	181	135	13	40	19	8	54
	Abnormal	13	40	19	8	54	13	40	19	8	54
	Total	189	189	189	189	189	26	80	38	16	108
Numerous-Traces	Normal	617	502	640	667	434	68	183	45	18	251
	Abnormal	68	183	45	18	251	68	183	45	18	251
	Total	685	685	685	685	685	136	366	90	36	502

established at three minutes. Among the best networks automatically obtained, the one with the smallest number of false negative classifications was chosen basing on the confusion matrix. The whole data set was divided into three subsets: learning, validating and testing. The classes' sizes were equal in every subset during the Equal-Approach. In Real-Approach the ratio of classes' sizes was constant for each subset and its value corresponded to the applied health attribute described by the NN output value.

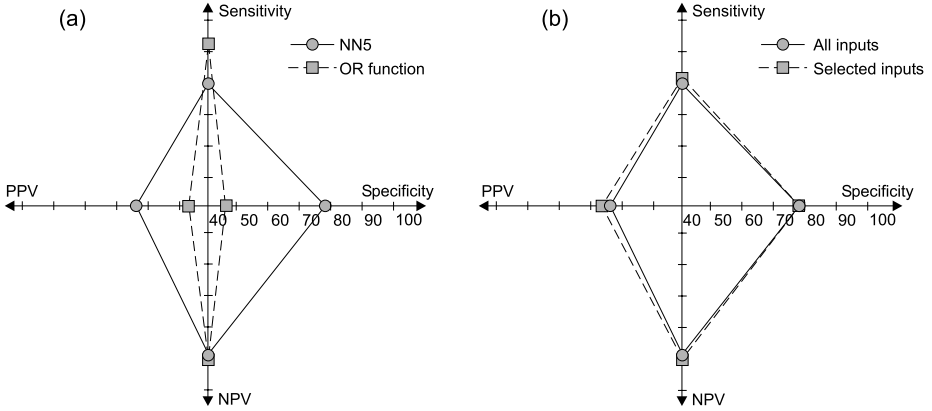
### 3 Results

The basal parameters describing effectiveness of the developed neural network are the classification quality and the classification error computed separately for each data subset. The first one is determined as a percentage of correct classifications, while the error is calculated as a root mean squared of the differences between the target and actual output values. Table 3 summarizes the results obtained for networks NN1÷NN5. The centres of neurons were determined by K-Means algorithm, the radii by K-Nearest Neighbours algorithm. The networks were learned on Real-Approach (Numerous-Traces) data set and with the all 21 input variables.

Each network had very similar values in each subset, which is very important, because it guarantees good ability of network's generalization. The lowest quality

**Table 3.** Parameters describing the classification features of the designed networks

Network	Learning		Validating		Testing	
	Quality [%]	Error	Quality [%]	Error	Quality [%]	Error
NN1	87	0.21	78	0.28	84	0.26
NN2	82	0.35	80	0.36	80	0.37
NN3	76	0.22	74	0.24	78	0.27
NN4	48	0.15	49	0.17	54	0.17
NN5	83	0.35	81	0.38	79	0.38



**Fig. 3.** Comparison of prognostic indices for NN5 network with OR function of NN1÷NN4 networks' outputs (a), and the influence of the reduced set of input variables on prognostic indices' values for NN5 network (b)

was for NN4 network, and it is connected with the fact, that the number of traces for abnormal value of A4 attribute was the lowest – only 18 (3 %) cases (Tab. 2). On the other hand, the lowest error values for NN4 network were observed.

The usefulness of particular networks was expressed using well known in medicine set of prognostic indices: Sensitivity (Sens), Specificity (Spec), Positive Predictive Value (PPV) and Negative Predictive Value (NPV) [[10]]. These indices were determined basing on confusion matrix calculated for the testing subset. The area limited by the values of prognostic indices (Fig. 3) was used to define the overall index OI. The weight in this index ensures, that the Sens plays the most important role. The OI formula is  $(2 * Sens + NPV) * (Spec + PPV) / 600$ .

From a clinical point of view, the number of false negative cases should be minimized, because they have more serious consequences than the false positive ones. Thus, the Sensitivity is a very important measure which should reach a high value. The same recommendation concerns the NPV index, which also relates to the number of false negative cases. However, it is much easy to achieve its high value because in clinical practice the number of true negative cases is much higher than the number of false negative ones.

Comparison of results of network NN5 designed for prediction of the fetal outcome, with the classification efficiency using the function OR on outputs of four individual networks designed separately, is shown in Fig. 3a. The network NN5 provided more stable values for all the four prognostic indices in comparison to the OR function of outputs. The increase of Sens and NPV in the OR function was accompanied by much larger decrease of Spec and PPV, and the OI index decreased by about 25% (Tab. 4). This caused that the NN5 network was chosen for further research.

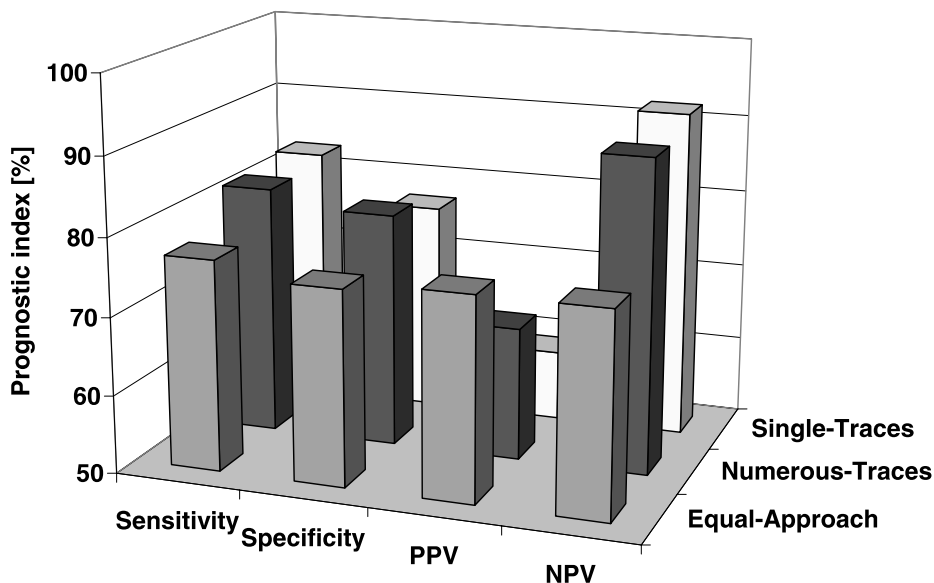


Fig. 4. The influence of various data grouping on values of the prognostic indices

Table 4. Summary of prognostic indices values presented in Fig. 3 and 4

Prognostic index	OR function		NN5		
	21 inputs		7 inputs		
	Numerous	Real	Numerous	Equal	Single
	Real	Real	Real	Equal	Real
	1	2	3	4	5
Sensitivity	93	79	82	77	83
Specificity	48	79	80	75	77
PPV	47	65	67	76	59
NPV	93	88	90	76	92
OI	0.44 (-25%)*	0.59	0.62 (+5%)*	0.58 (-6%)#	0.58 (-6%)#

(\*)\* - the indirect change of index OI in relation its value in the column No 2  
 (#)# - the indirect change of index OI in relation its value in the column No 3

Rejection of some input variables didn't decrease values of prognostic indices as it can be expected (Fig. 3b), but even insignificantly increased them (OI greater by about 5 %, Tab. 4). Creating a new set of 7 input variables was considered as favourable.

## 4 Conclusion

In presented work, a number of experiments concerning modifications of the network's inputs and outputs, as well as the data set, were done. On the basis

of the results, the best neural networks learned on fetal outcome was chosen, whose task was to classify CTG traces as corresponding to normal or abnormal fetal state.

After the comparison of the OR function of results of four networks designed for prediction of four particular attributes describing fetal outcome, with the results obtained from the network predicting the final fetal outcome assessed by expert, the second approach was chosen. From various ways of data grouping the one with several CTG traces for one patient and with disproportion between classes' sizes was chosen, which is similar to clinical practice. Finally it was proved, that rejection of some input variables can provide better results.

The presented methodology is quite similar to [3], where large thirty-element set of input parameters of quantitative description of CTG traces was used. One patient was associated with only one trace. Abnormal fetal outcome was defined in the same way as in this work. The final fetal outcome classification results were as follows: Sens (73%), Spec (94%), PPV (72%), NPV (94%).

Results obtained by us are slightly worse (the OI = 0.66 worse by about 6%) in comparison to [3]. It should be pointed out, that authors [3] used much more input variables and different network type (MLP). However, we obtained significant improvement of sensitivity thanks to application of the selection of the best network, which was based on the minimum number of false negative cases.

The artificial neural network approach for the prediction of fetal outcome based on parameters of quantitative description of CTG traces seems to be very attractive. Details analysis of research material had a strong influence on the research procedure. During various experiments the data set content and structure, as well as inputs and outputs representation, were changed. The results of experiments gave answers to many questions, which occur during development of conclusion generation systems based on neural networks for medical applications.

## Acknowledgement

This work was supported in part by the Ministry of Sciences and Higher Education resources in 2007-2009 under Research Project R13 028 02.

## References

1. Dalton K.J.: Neuro-computing: a medical overview of neural networks, with particular reference to obstetrics and gynaecology in "A critical appraisal of fetal surveillance", ed: H. Geijn, Elsevier, 1994, 625-632.
2. Jezewski J., Wrobel J., Horoba K.: Computerized perinatal database for retrospective qualitative assessment of CTG traces in „Current Perspectives in Healthcare Computing”, ed: B. Richards, BTHC, 1996, 187-196.
3. Arduini D., Giannini F., Magenes G.: Intrapartum surveillance: computer cardiotocography, Proc. of 5th World Congress of Perinatal Medicine, 2001, 1217-1223.



4. Jezewski J., Wrobel J., Horoba K., Kupka T., Matonia A.: Centralised fetal monitoring system with hardware-based data flow control, Proc. of III International Conference MEDSIP, Glasgow, VII 2006, 51–54.
5. Jezewski M., Henzel N., Wrobel J., Labaj P., Horoba K., Matonia A.: Application of neural networks for prediction of fetal outcome, Journal of Medical Informatics and Technologies, 2006, Vol.10, 127-132.
6. Jezewski M., Wrobel J., Labaj P., Leski J., Henzel N., Horoba K., Jezewski J.: Some practical remarks on neural networks approach to fetal cardiocograms classification, 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, /in print/.
7. Labaj P., Jezewski M., Matonia A., Kupka T., Jezewski J., Gacek A.: New approach to quantitative description of deceleration of fetal heart rate for the patterns classification, 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, /in print/.
8. Magenes G., Signorini M.G., Arduini D.: Classification of cardiocographic records by neural networks, Proc.of the IEEE International Joint Conference on Neural Networks, 2000, Vol. 3, 637–641.
9. Magenes G., Signorini M.G., Sassi R.: Multiparametric analysis of fetal heart rate: comparison of neural and statistical classifiers, Proc. of IX Conference on Medical Engineering and Computing, 2001, 360–363.
10. Sikora J.: Digital analysis of cardiocographic traces for clinical fetal outcome prediction, Clinical Perinatology and Gynaecology, Supplement 21, 2001, 57-88.

---

# Determining Weights of Symptoms in a Diagnostic Inference

Ewa Straszecka

Institute of Electronics, Silesian University of Technology  
16 Akademicka St, 44-100 Gliwice, Poland  
ewa.straszecka@polsl.pl

**Summary.** The paper focuses on modeling weights of symptoms in medical diagnosis. A model of diagnostic inference is proposed in the framework of the Dempster-Shafer theory extended for fuzzy focal elements. The basic probability assignment defined in this theory estimates weights of symptoms. Two such assignments - determined for 'easy' diagnostic cases and 'difficult' cases, respectively, are calculated for training data, and next combined. Resulting values of the basic probability assigned to focal elements make knowledge about the diagnosis. An algorithm of tuning weights by means of training data is suggested and tested for a medical database from the Internet.

## 1 Introduction

Diversity of symptoms has to be analyzed during medical diagnosis. Results of laboratory tests are numbers and they provide crisp (precise) information. On the other hand, symptoms from an interview or primary examination are often linguistically formulated. All symptoms need to be estimated according to their importance in the diagnosis. Weights of the symptoms can be determined by an expert or can be found on the basis of training data. When a model of diagnosis is based rather on data mining than on representation of human knowledge, the weights can be approximated by probability values. Then, a problem arises how to estimate probabilities of dependent symptoms. The problem is even more complex if symptoms differ in nature, i.e. some are numerical and the other are linguistic. The common way of estimation is to consider exclusively 'yes/no' symptoms. It means that only presence or absence of a symptom is analyzed, regardless of its imprecision. Still, exacerbation of a symptom is important in inference and should be represented during reasoning. The present paper suggests modeling weights of symptoms by the basic probability assignment defined in the Dempster-Shafer theory of inference [1]. An extension of the theory for fuzzy focal elements [5] makes it possible to consider both certainty and precision of a symptom. Values of the basic probability assignment are calculated for single symptoms as well as for their collections. The collections may gather numerical and linguistic symptoms. An algorithm of the assignment calculation is proposed. In the algorithm a combination of two assignments found from training data is used. The first assignment is determined for 'easy' training cases, while the second for the cases difficult for classification. In consequence, weights of symptoms

can be tuned for better performance of classification. An other outcome of the algorithm is a set of diagnostic rules relating symptoms to diagnoses.

## 2 Methods

### 2.1 The Dempster-Shafer Theory

In the Dempster-Shafer theory (DST) the basic probability assignment (bpa), denoted as  $m$ , is defined in the following way [1]:

$$m(f) = 0, \sum_{a \in A} m(a) = 1. \tag{1}$$

where  $f$  stands for the false focal element, and  $A$  is a set of all focal elements  $a$ . Notation in (1) and in the following formulas is identical or similar to that used in [1], as it seems to be the most convenient for modeling of medical diagnosis. In a model of medical inference the focal element  $a$  describes a symptom ( $a_i = \{x\}$ ) or a collection of several symptoms ( $a_j = \{x_1, \dots, x_k\}$ ). The former will be called the single focal element while in the latter - the complex focal element. When two sources provide information about symptoms (i.e. focal elements), two bpas are defined, which can be combined according to the rule [1]:

$$m(a) = m_1 \oplus m_2(a) = \frac{\sum_{a_i \cap a_j = a} m_1(a_i)m_2(a_j)}{\sum_{a_i \cap a_j \neq f} m_1(a_i)m_2(a_j)}. \tag{2}$$

The focal element can describe not only a symptom, but also a diagnosis. While the bpa is known, belief and plausibility measures [1] can be determined:

$$Bel(c) = \sum_{(a \Rightarrow c)=t} m(a), \tag{3}$$

$$Pl(c) = \sum_{(a \Rightarrow c) \neq f} m(a), \tag{4}$$

where  $t$  stands for truth. Focal element  $c$  can be regarded as a conclusion and element  $a$  as a premise of a diagnostic rule, since  $c$  results from  $a$ . Still, the right arrow  $\Rightarrow$  in (3) and (4) should be understood as an assignment rather than an implication. The focal element  $a$  includes symptom(s)  $\{x_i\}$  that support(s) diagnosis  $c$ . Unlike symptoms, focal elements  $c$  are crisp and single. According to this interpretation, the  $[Bel(c), Pl(c)]$  interval corresponds to the certainty of the diagnosis. Thus, (3) and (4) are suitable for a representation of belief and plausibility of a diagnostic hypothesis. When several hypotheses have to be analyzed during an inference, their belief measures can be compared. They should be found for maximal plausibility of the hypotheses that guarantees that all possible information has been used to find a conclusion. Still, more attention should be paid to  $Bel$ , i.e. the smaller measure, as conclusions in this field are drawn very cautiously. Thus, the hypothesis with the greatest  $Bel$  value is the final conclusion. If the greatest value occurs with several hypotheses, the final conclusion cannot be determined.

### 2.2 Fuzzy Focal Elements

The focal element  $a$  is either a numerical or a linguistic symptom or includes several symptoms. Thus, the value of  $bpa$  is the weight of any kind of a symptom or a collection of symptoms in the diagnosis. The problem of imprecision of a symptom, particularly in case of its linguistic formulation, can be solved if a fuzzy set is used for its description. The fuzzy set is usually defined on a domain that is a common scale for the symptom. For instance, 'little loss of weight' is determined by a fuzzy set on the domain of kilograms. Yet, the same is possible for an assumptive scale, e.g. for the VAS scale of pain. A fuzzy membership function is also profitable to approximate norms of laboratory tests, particularly when they change according to a laboratory procedure (or other circumstances). All in all, when a single focal element is defined by a fuzzy set, an imprecision level can be determined:

$$\eta_i = \max_{x_i} \min(\mu_l(x_i), \mu_l^*(x_i)), \tag{5}$$

where  $\mu_l(x_i)$  denotes the fuzzy membership function that in the  $l$ -th diagnosis characterizes the  $i$ -th symptom. This membership function is an element of knowledge about diagnosis and can be both found from training data or defined by an expert. In the same time,  $\mu_l^*(x_i)$  denotes an evidence, which is an input for the diagnostic inference. It comes from test data. The fuzzy sets  $\mu_l(x_i)$ ,  $\mu_l^*(x_i)$  are generalizations of symptoms. Firstly, they can be fuzzy sets. Secondly, when  $x_i$  is a numerical value, then  $\mu_l, \mu_l^*$  are singletons on a numerical domain. The 'yes/no' symptoms can be also represented by singletons on the  $\{0, 1\}$  domain. Knowledge ( $\mu_l$ ) and an input of an inference ( $\mu_l^*$ ) may differ in nature. For instance norms of a laboratory test may be fuzzy, while the result of the test for a patient is a singleton.

During the belief calculation (3) for complex focal elements  $a$  the imprecision level is found as minimum of the levels for symptoms included in the collection, while for the plausibility (4) as maximum of the levels. Then:

$$Bel(c) = \sum_{\substack{a \Rightarrow c \\ \min_a(\eta_a) \geq \eta_T}} m(a), \tag{6}$$

$$Pl(c) = \sum_{\substack{a \Rightarrow c \\ \max_a(\eta_a) \geq \eta_T}} m(a), \tag{7}$$

where  $\eta_T$  stands for a threshold assumed for the inference. Such an interpretation of imprecision levels follows classical definitions of  $Bel$  and  $Pl$ . The focal element  $a$  involves hypothesis  $c$  if its imprecision level is greater or equal to the threshold value.

### 2.3 Calculation of the Basic Probability Assignment

Let us assume that a data set comprising symptoms of patients and their diagnoses is the only source of information about a disease. It is inevitable that the

data include at least two kinds of patients: with diagnosis  $D_1$  (e.g. a disease) and  $D_2$  (e.g. health), otherwise a differentiation between the diagnostic hypotheses would not be possible. Yet, the number of hypotheses can be greater. An algorithm that uses the DST extended for fuzzy focal elements to model weights of symptoms is provided as a distinct paragraph of this section. The set of data should be divided for subsets according to  $N$  diagnoses. Membership functions and bpas are individually determined for each diagnosis. Number of training cases of the  $l - th$  diagnosis is  $n_l$ . Membership functions for numerical variables can be found as trapezoid functions based on some statistical parameters. The method of membership function design for the present algorithm is described in details in [5]. Membership functions for non-numerical variables can be designed only when more information than 'yes/no' is provided about presence of symptoms. Unfortunately, this hardly happens in available databases, but it seems to be quite possible in future.

Statistical procedures may be used to find dependencies among variables in training data. Single focal elements are defined for all variables. They are supplemented by complex focal elements defined for dependent variables. For the complete set of focal elements regarding individual diagnostic hypothesis the bpa is found. The bpa values are calculated as normalized frequency of occurrence of training cases with each focal element. Next, the same training cases are diagnosed by calculating belief in each hypothesis: e.g. the belief in the  $D_l$  diagnostic hypothesis for the  $i - th$  set of data is denoted by  $Bel(\bar{x}_i, D_l)$ ,  $l = 1, \dots, N$ . The threshold  $\eta_T$  has to be assumed. Belief values calculated for training cases, are next compared. When the greatest value of the belief is single and it indicates the same diagnosis as originally assigned to the training case, the case is correctly classified. Majority of training cases is usually correctly classified. They are comprised in a 'correct cases' set. Yet, some wrongly classified cases also occur and they create a 'wrong cases' set. Now, it is possible to calculate two bpas for each diagnosis in the present iteration of the algorithm: bpa of the 'correct cases' ( $m_{1l}^{(k)}$ ) and 'wrong cases' ( $m_{2l}^{(k)}$ ). Their combination (2) results in the bpa that is used for classification in the next iteration. Membership functions of the two subsets of training data should be also combined. To this end, minimum of adequate functions defined for the same domain is found:

$$\mu_l^{(k)}(x_i) = \min \left( \mu_{1l}^{(k)}(x_i), \mu_{2l}^{(k)}(x_i) \right). \tag{8}$$

Thus, bpa and membership functions are determined. The process of the bpa tuning can now be repeated by an iterative classification of training data for 'correct' and 'wrong' cases. In the  $k - th$  iteration the global error  $\epsilon^{(k)}$  and classification errors for diagnoses  $\epsilon_l^{(k)}$  are calculated:

$$\epsilon^{(k)} = \frac{\sum_{l=1}^N n_{lwrong}}{\sum_{l=1}^N n_l}, \quad \epsilon_l^{(k)} = \frac{n_{lwrong}}{n_l}, \tag{9}$$

*Algorithm of tuning the basic probability assignment by means of training data*

Assumptions:  $K$  - number of iterations,  $\eta_T$  - threshold for inference

1. Training data:  $\{\bar{x}_i, D_i\}$ ,  $i = 1, \dots, n_l$ ,  $l = 1, \dots, N$ ;
2. Determination of membership functions for fuzzy focal elements;
3. Determination of focal elements through statistical investigations;
4.  $k = 1$ , calculation of the bpas  $m_l^{(k)}$ ,  $l = 1, \dots, N$ ;
5. Calculation of the  $Bel(\bar{x}_i, D_l)$ ,  $i = 1, \dots, n_l$ ,  $l = 1, \dots, N$  for the  $\bar{x}_i$  case and the  $\eta_T$ , (5)-(7);
6. Classification of training data by comparing  $Bel(\bar{x}_i, D_l)$ ,  $l = 1, \dots, N$ . If the maximal  $Bel$  value is single and  $Bel(\bar{x}_i, D_l^*) = \max_l(Bel(\bar{x}_i, D_l)) \Rightarrow D_l^* = D_i$  then the  $i$ -th case is 'correct', otherwise the case is 'wrong';
7. Calculation of errors  $\epsilon^{(k)}$ ,  $\epsilon_l^{(k)}$ ,  $l = 1, \dots, N$ , (9);
8. Calculation of the bpa for 'correct cases'  $m_{1l}^{(k)}$  and 'wrong cases'  $m_{2l}^{(k)}$ ,  $l = 1, \dots, N$ ;
9. Combining the bpas  $m_l^{(k+1)} = m_{1l}^{(k)} \oplus m_{2l}^{(k)}$ ,  $l = 1, \dots, N$  according to (2) and membership functions according to (8);
10.  $k = k + 1$ , repetition of steps (5-10) until  $\epsilon^{(k)} = 0$  or  $k = K$ .

The bpa that results in the minimal  $\epsilon_l^{(k)}$  is the appropriate bpa for the  $l$ -th diagnosis. Then test data can be classified by means of the bpa and the adequate membership functions.

where  $n_{wrong}$  is the number of 'wrong' cases for the  $l$ -th diagnosis. Iterations proceed until  $\epsilon^{(k)} = 0$  or the maximum number of iterations ( $K$ ) is reached. As a result weights that are at first roughly estimated for the majority of cases are next modified by cases that are the most difficult for classification. In this way the weights of symptoms are tuned individually for each diagnosis. It is worth to notice that bpas obtained for 'correct' and 'wrong' cases are not contradictory. The combination of the bpas aims at emphasizing of weights of some focal elements to improve performance of the algorithm. The belief can be an important measure of certainty of diagnosis for a user (a physician) if a diagnosis support tool would base on the algorithm.

### 3 Tests and Results

#### 3.1 Database

The algorithm from the subsection 2.3 has been tested for the database <ftp.ics.uci.edu/pub/machine-learning-databases/thyroid-disease>, files *ann.\**.

The data include 6 numerical and 15 two-valued (1 and 0) variables with 3 diagnoses assigned. Training/test files have the following number of cases:

$D_1$  - 191/177,  $D_2$  - 93/73,  $D_3$  - 3488/3178. Authors of the database demand 92% correctness of classification. References methods [2], [3] that introduce neural networks and genetic algorithms, report an error smaller than 2%. Still, the authors of the papers admit that such a classification lack generalization. Thus, the present algorithm aimed at the error of classification smaller than 8%, but in the same time in a classification procedure that indicates general rules.

### 3.2 Calculations

Single focal elements have been defined for all 21 variables. Complex focal elements have comprised variables for which dependence have been statistically confirmed by the  $\chi^2$  test and  $Q$ -Kendall measure [6]. Each complex focal element has included at most 5 symptoms. Complex elements with numerous symptoms (up to 15) as well as combinations of statistically independent variables have been also tested. Yet, results have been disappointing. Collections of dependent variables have been investigated to find focal elements with greater than 0 frequency of occurrence among training data. Eventually, 295 focal elements have been created. During each run of the algorithm 3 trapezoidal membership functions corresponding to the diagnoses have been defined for every numerical variable. Thanks to a particular 'wide' shape of membership functions and small number of numerical variables in comparison to two-valued variables a threshold for the imprecision level (5) has been of minor importance for the present considerations. That is why an analysis of the threshold has been neglected. The threshold has been assumed as  $\eta_T \geq 0.5$ . More details on this topic can be found in [4]. Several modifications of the final bpa, like e.g. increase of contrast among its values, have been tried. The best results have been obtained when each focal element has been assigned exclusively to one diagnosis - the one that has been indicated by its maximum weight. Obviously, after determining the maximum weights, the bpas have been again normalized to fulfill (1). During calculations the minimum error for a single diagnosis has been searched. Different numbers of iterations have been necessary to find such a minimum: 29 for  $D_2$ , 66 for  $D_3$  and 87 for  $D_1$ . Thus, much less than 100 iterations are sufficient to find acceptable weights for symptoms. This seems to be a considerable advantage in comparison to neural networks and genetic algorithms [2],[3].

### 3.3 Results and Discussion

For the training data the following minimal errors have been obtained for the diagnoses: 1.8% for  $D_1$ , 2.6% for  $D_2$  and 3.8% for  $D_3$ . The global error, i.e. the error calculated for all cases included in one training set has been equal 3.7%. The bpas that resulted in these errors have been next used to classify test data. Again the errors have been much smaller than demanded by the 8% criterion of performance. They have been: 0.0% for  $D_1$ , 5.1% for  $D_2$  and 4.0% for  $D_3$ .

The global error has equaled 4.0%. The algorithm correctly classifies the data, and determines focal elements as well as their weights in inference. Thus, we obtain not only a classification tool, but also a method to get knowledge from training data. An expert can easily verify weights of symptoms, and so inference rules can be built. Moreover, it is possible to combine expert's knowledge with the bpa obtained from data in the same way it is done for two bpas. It is even not necessary that focal elements are identically defined. If we assume that focal elements are premises of rules, conclusions of the rules are diagnoses and the bpa values are weights of rules, we obtain a knowledge base for diagnosis support.

## 4 Conclusions

The algorithm that is proposed in the paper makes it possible to estimate a weight of a symptom or a collection of symptoms in the diagnosis, regardless symptom's nature. The symptom can be represented by the fuzzy set. In this way both certainty and precision of symptoms are considered. The weights are modeled by the basic probability assignment defined in the Dempster-Shafer theory. The probability assignment neglects dependence of symptoms that reduces complexity of calculation of probability values. During training two basic probability assignments are calculated, respectively determined for the cases that are easy and difficult for classification. Afterward, the assignments are combined. As a result weights that are at first roughly estimated for majority of cases are next modified by the most specific cases. Such a method has a better performance than a single probability assignment, because specific cases are usually less numerous and do not counterbalance the average cases in the simple training process. The combination of the assignments emphasizes weights of some focal elements and in this way improves classification. Another advantage of the method is that focal elements, related diagnoses and assigned weights make a knowledge base for the diagnosis support. An expert can verify and/or supplement it by his own estimation. This is not possible with the methods of classifications that use neural networks or genetic algorithms. The belief is an important measure of certainty of diagnosis for the possible user (a physician) of the diagnosis support tool. The algorithm has been tested on a medical database of thyroid gland diseases. The classification error has been smaller than that demanded for the database. Therefore, the algorithm seems to be worth considering for solving joint problems of classification and data-mining.

## References

1. Kacprzyk J, Fedrizzi M (1994) *Advances in Dempster-Shafer Theory of Evidence*, Wiley, New York.
2. Schiffman W, Joost M, Werner R (1992) *Synthesis and performance analysis of multilayer neural network architectures*, ftp archive.cis.ohio-state.edu, technical report



3. Schiffman W, Joost M, Werner R (1994) Optimization of Backpropagation Algorithm for training multilayer perceptrons, ftp archive.cis.ohio-state.edu, technical report
4. Straszecka E (2003) An interpretation of focal elements as fuzzy sets, *Int. J. of Intelligent Systems* 18:821–835.
5. Straszecka E (2006) Combining uncertainty and imprecision in models of medical diagnosis, *Information Sciences*, 176:3026–3059
6. Tadeusiewicz R, Izvorski A, Majewski J (1993) *Biometria*, Wydawnictwa AGH, Krakow

---

# Hough Transform, (1+1) Search Strategy and Watershed Algorithm in Segmentation of Cytological Images

Maciej Hrebien, Józef Korbicz, and Andrzej Obuchowicz

Institute of Control and Computation Engineering  
University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, Poland  
{m.hrebien, j.korbicz, a.obuchowicz}@issi.uz.zgora.pl

**Summary.** This paper describes an early stage of cytological image recognition and presents a proposition of a hybrid segmentation method. The analysis includes the Hough transform in conjunction with the watershed algorithm. One can also find here a short description of image pre-processing and an automatic nuclei localization mechanism used in our approach. Preliminary experimental results collected on a hand-prepared benchmark database are also presented with discussion of common errors and possible future problems.

## 1 Introduction

The construction of a fully automatic cancer diagnosis system has become a challenging task. In last decade we have been observing a very dynamic growth in number of research works conducted in this area not only by university centers but also by commercial institutions [2]. Because breast cancer is becoming the most common form of cancer disease of today's female population, much attention of the present-day researchers is directed to this issue. The attention covers not only curing the external effects of the disease but also its fast detection in its early stadium.

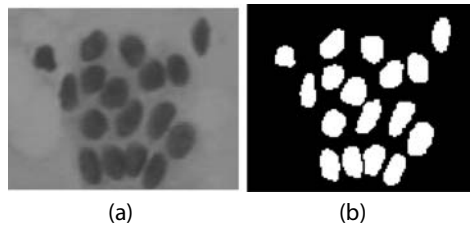
The nucleus of the cell is the place where breast cancer malignancy can be observed. Therefore, it is crucial for any camera-based automatic diagnosis system to separate the cells and their nuclei from the rest of the image content. Until now many segmentation methods have been proposed [3, 8, 9] but, unfortunately, each of them introduces different kinds of additional problems and usually works in practice under given assumptions and/or needs the end-user's interaction/co-operation. Since many cytological projects assume rather full automation and real-time operation with high degree of efficacy, a method free of drawbacks of the already known approaches has to be constructed.

In this paper a hybrid method of cytological image segmentation is presented, that is the Hough transform in conjunction with the watershed algorithm. One can also find here a short description of image pre-processing and fully automatic nuclei localization mechanisms used in our approach.

## 2 Problem Formulation

Mathematical formulation of the segmentation process is very difficult because it is a poorly conditioned problem. Thus we give here only an informal definition of the problem we have to face.

What we have on input is a cytological material obtained using the Fine Needle Biopsy technique and imaged with a *Sony CCD Iris* camera mounted atop of an *Axiophot* microscope. The material comes from female patients of Zielona Góra's *Onkomed* medical center [7]. The image itself is coded using the RGB colorspace and is not subject to any kind of lossy compression (araw color bitmap format). The size of the image equals  $704 \times 576$  pixels.



**Fig. 1.** Exemplary fragment of: (a) cytological image, (b) appropriate segmentation mask

What we expect on output is a binary segmentation mask with one pixel separation rule which will allow more robust morphometric parameters estimation in our future work. Additionally, the algorithm should be insensible to colors of contrasting pigments used for the preparation of the cytological material (see an example in Fig. 1).

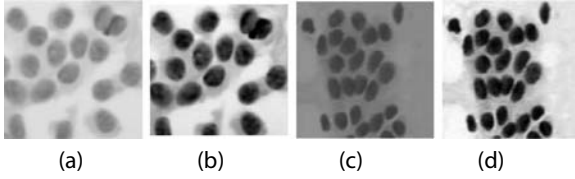
## 3 Image Pre-segmentation

### 3.1 Pre-processing

The color components of an image do not carry as important information as the luminosity does, so they can be removed to reduce processing complexity in stages that require only, e.g., gradient estimations. An RGB color image can be converted to greyscale by removing blue and red chrominance components from the image defined in the YCbCr color space leaving only the luminosity one [8].

Since the majority of images we deal with have low contrast, an enhancement technique is needed to improve their quality. In our approach we use simple histogram processing with a linear transform of image's levels of intensities, namely the cumulated sum approach [9] with 1% saturation at low and high intensities of the input image.

The contrast correction operation is conducted for each color channel separately resulting in an image being better defined for later stages of the presented hybrid segmentation method (see an example in Fig. 2).



**Fig. 2.** Exemplary fragments of cytological images (a, c) and their contrast enhanced equivalents (b, d)

### 3.2 Background of the Algorithm

If we look closely at the nuclei we have to segment, we notice that they all have an circular shape (Fig. 2a). Most of them resemble an ellipse but, unfortunately, detection of ellipse which is described by two parameters  $a$  and  $b$  ( $x = a \cos \alpha$ ,  $y = b \sin \alpha$ ) and which can be additionally rotated is computationally expensive. The shape of the ellipse can be approximated by a given number of circles. The detection of circles is much simpler in the sense of required computations because we have only one parameter, which is the radius  $R$ . These observations and simplifications constitute grounding for a fast nucleus pre-segmentation algorithm – in our approach we try to find such circles with different radii in a given feature space.

### 3.3 Circles Detection

The Hough transform [11, 12] can be easily adopted for the purpose of circle detection. The transform in the continuous space could be defined by:

$$HT(x_0, y_0, R) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta\left((x - x_0)^2 + (y - y_0)^2 - R^2\right) dx dy, \quad (1)$$

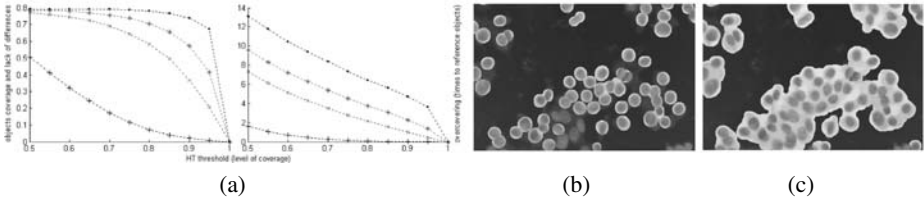
where  $f$  is any continuous function defining the space in which the circles will be searched for and  $\delta$  is Dirac's delta (infinite impulse answer at zero), which defines integration only over the circle. The  $HT$  plays the role of an accumulator which accumulates the levels of feature image  $g$  similarity to the circle placed at the  $(x_0, y_0)$  position and defined by the radius  $R$ .

The feature space  $g$  can be created by many different ways. In our approach we use a gradient image as the feature indicating nucleus occurrence or absence in a given fragment of the cytological image. The gradient image is a saturated sum of gradients estimated in eight directions on the greyscale image prepared in the pre-processing stage. The base gradients can be calculated using, e.g., Prewitt's, Sobel's mask methods or their heavy or light versions [3, 10].

### 3.4 Final Actions

Thresholding the values in the accumulator by a given  $\theta$  value can lead us to a very good pre-segmentation mechanism with the lower threshold strategy

(see, for instance, Fig. 3bc). Since the threshold value strongly depends on the database and used feature image  $g$  (Fig. 3), the method can only be used as a pre-segmentation stage. Smaller value of the threshold causes fast removal of non-important information from the background what can constitute a base for more sophisticated and going into details algorithms.



**Fig. 3.** Influence of the threshold value on the objects’s cover and the lack of differences (left) and overcovering (right) for the Prewitt (×), Sobel (\*), heavy (•) and light (+) base gradient masks (experiments performed on a randomly selected 346 element Zielona Góra *Onkomed* [7] cytological benchmark database for radii in the 4-21 range) (a); exemplary results of the pre-segmentation stage for two different threshold strategies: (b) high and (c) low

## 4 Image Segmentation

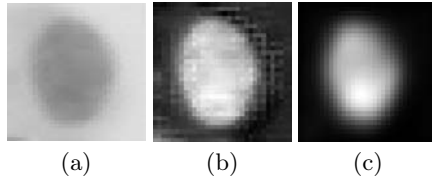
### 4.1 Terrain Modeling

The results obtained from the pre-segmentation stage can lead us to the estimation of an average background color. Such information can be used to model the nuclei as a color distance between the background and the objects what fulfills the requirements of the lack of any color dependency in the imaged material (the color of contrasting pigments may change in the future). In our research we tried few distance metrics: Manhattan’s, Chebyshev’s, the absolute Hue value from the HSV colorspace, but the Euclidian one gives us visually the best results (Fig. 4ab):

$$D_{euclid} = \sqrt{(I_R - B_R)^2 + (I_G - B_G)^2 + (I_B - B_B)^2}, \tag{2}$$

where  $B$  is the average background color estimated for the  $I$  input image.

Since the modeling distance can vary in the local neighborhood (see Fig. 4b) mostly because of camera sensor simplifications, a smoothing technique is needed to reconstruct the nuclei shape. The smoothing operation in our approach relies on the fact that this sort of 2D signal can be modeled as a sum of sinusoids [6] with defined amplitudes, phase shifts and frequencies. Cutting all low amplitude frequencies off (leaving only a few significant ones with the highest amplitude) will result in a signal deprived of our problematic local noise effect (Fig. 4c). The frequency spectrum is determined using the discrete Fourier transform and after the frequency cut off the signal is reconstructed using its inverse. What we finally achieve is a three-dimensional modeled terrain where hills correspond to nuclei.



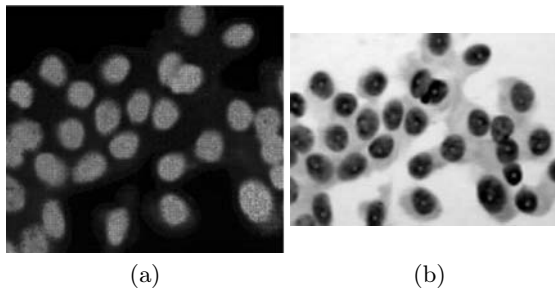
**Fig. 4.** Exemplary fragment of: (a) cytological image, (b) Euclidian distance to the mean background color, (c) smoothed out version of (b)

## 4.2 Nuclei Localization

The localization of objects on a modeled map of nuclei can be performed locally using various methods. In our approach we have chosen an evolutionary (1+1) search strategy [1] mostly because it is simple, quite fast despite appearances, can be easily parallelized due to its nature and it settles very well in local extremas, which is very important in our case.

The final segmentation algorithm used forced us to create two population of individuals. The first population localizes the background. Specimens are moved with a constant movement step ( $R = 1$ ) preferably to the places with a smaller density of population to maximize background coverage. The second population localizes the nuclei. Specimens are moved with a decreasing movement step ( $r^t = R_{max} \left( \frac{1}{R_{max}} \right)^{\frac{t}{t_{max}}}$ ) to group very fast the population near local extremas in the first few epochs and to finally work on details in the ending ones. The movement of individuals is preferred to be directed towards the places with a higher population density to create the effect of nuclei localization.

The mutation is defined as  $Y_i^t = X_i^t + r^t N_i(0, 1)$  and the fitness function  $\phi$  calculates the average *height* of the terrain in a given position including nearest neighborhood defined by the smallest radius detected by the Hough transform in the pre-segmentation stage. Such a definition of the fitness function avoids a possible split of the population, localized near a nucleus with multimodal character of its shape, giving only one marker for a nucleus (Fig. 5b).



**Fig. 5.** Exemplary localization: (a) screenshot after 8 epochs, (b) final result (localization points are marked with red asterisks)

Finally, the nucleus is localized in the place where the density of the population searching for hilltops in the modeled terrain is locally maximal. As we have mentioned earlier, the method is fast and just a few epochs are needed to observe a visible progress in nuclei localization and background coverage (Fig. 5a).

### 4.3 Building Watersheds

The watershed segmentation algorithm is inspired by natural observations, such as a rainy day in the mountains [3, 8, 9]. A given image can be defined as a terrain on which nuclei correspond to valleys (upside down the terrain modeled in previous steps). The terrain is flooded by rainwater and arising puddles are starting to turn into basins. When the water from one basin begins to pour away to another, a separating watershed is created.

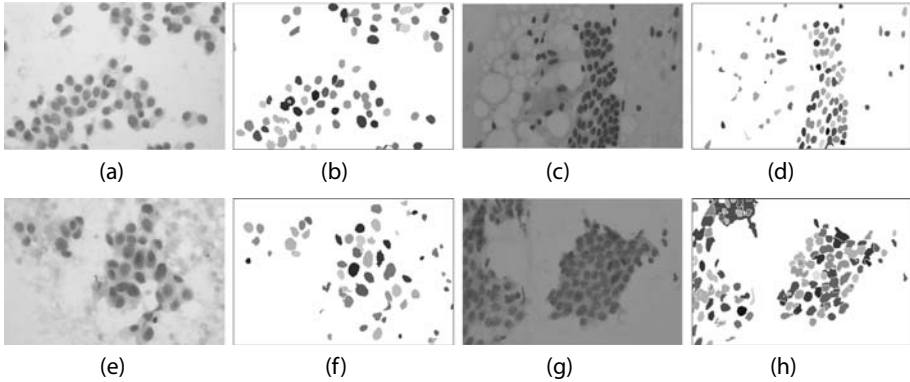
The flooding operation has to be stopped when the water level reaches a given threshold. The threshold should preferably be placed somewhere in the middle between the background and a nucleus localization point. In our approach the nuclei are flooded to the half of the altitude between nucleus localization point and the average height of the background in the local neighborhood. Since the images we have to deal with are spot illuminated during the imaging operation (resulting in a modeled terrain higher in the center of the image and much lower in the corners), this mechanism protects the basins from being overflowed and, in consequence, nuclei being undersegmented.

### 4.4 Exemplary Results

Exemplary results of the presented hybrid segmentation method and common errors observed in our hand-prepared benchmark database can be divided into four classes:

- *class 1*: good quality images with only small irregularities and rarely generated subbasins (a basin in another basin) (Fig. 6ab),
- *class 2*: errors caused by fake circles created by spots of fat (Fig. 6cd),
- *class 3*: mixed nucleus types: red and purple in this case and those reds which are more purple than yellow (background) are also segmented, which is erroneous (Fig. 6ef),
- *class 4*: poor quality image with a bunch of nuclei glued together, which causes basin's overflowing and, in consequence, undersegmentation (Fig. 6gh).

The conducted experiments show that the watershed algorithm gives a 68.74%, on average, agreement with the hand-prepared templates using a simple XOR metric. Most errors are located at boundaries of nuclei where the average distance between the edges of segmented and reference objects is about 3.28 pixels. The XOR metric is underestimated as a consequence of not very high level of water flooding the modeled terrain, but the shape of the nuclei seems to be preserved, which is important in our future work – the estimation of morphometric parameters of segmented nuclei.



**Fig. 6.** Exemplary results of the watershed segmentation

## 5 Conclusions

The conducted preliminary experiments show that the Hough transform adopted for circle detection in the pre-segmentation stage, the (1+1) search strategy used for automatic nuclei localization and the watershed algorithm for the final segmentation stage can be effectively used for the segmentation of cytological images. Unfortunately, there are still some problems requiring further research.

The problem regarding fake circles created by spots of fat and unwanted effects it gives in the final output should be considered and eliminated in future work. Images with a mixed nucleus type still constitute a challenge because it seems to be impossible to detect only one type without the end-user's interaction and when there should not be any dependencies and assumptions concerning the color of contrasting pigments used to prepare cytological material. The proposal hybrid method should also be extended to perform better on poor quality images or a fast classifier [5] should be constructed to reject too poor (or even fake) inputs.

Summarizing, the presented solution is promising and gives a good base for our further research in the area of cytological image segmentation. Additionally, all preparation steps including pre-segmentation and the automatic nucleus localization stage can be reused with other segmentation algorithms which need such information (see, for example, [4]).

## References

1. J. Arabas: Lectures on Evolutionary Algorithms, WNT, 2004 (in Polish)
2. M. Kimmel, M. Lachowicz, A. Świerniak (Eds.): Cancer Growth and Progression, Mathematical Problems and Computer Simulations, Int. J. of Appl. Math. and Comput. Sci., Vol. 13, No. 3, Special Issue, 2003
3. R. Gonzalez, R. Woods: Digital Image Processing, Prentice Hall, 2002



4. M. Hrebień, P. Steć: Fine Needle Biopsy Material Segmentation with Hough Transform and Active Contouring Technique, *Journal of Medical Informatics & Technologies*, Vol. 10, pp. 25–34, 2006 (in print)
5. J. Łęski: A Fuzzy If-Then Rule-Based Nonlinear Classifier, *Int. Journal of Appl. Math. and Comput. Science*, Vol. 13, No. 2, pp. 215–223, 2003
6. V. Madisetti, D. Williams: *The Digital Signal Processing Handbook*, CRC, 1997
7. A. Marciniak, A. Obuchowicz, R. Monczak, M. Kołodziński: Cytomorphometry of Fine Needle Biopsy Material from the Breast Cancer, *Proc. of the 4th Int. Conf. on Comp. Recogn. Systems CORES'05*, *Adv. in Soft Computing*, pp. 603-609, Springer, 2005
8. W. Pratt: *Digital Image Processing*, John Wiley & Sons, 2001
9. J. Russ: *The Image Processing Handbook*, CRC Press, 1999
10. R. Tadeusiewicz: *Vision Systems of Industrial Robots*, WNT, 1992 (in Polish)
11. P. Toft: *The Radon Transform*, Ph.D. thesis, Tech. University of Denmark, 1996
12. W. Żorski: *Image Segmentation Methods Based on the Hough Transform*, Studio GiZ Warszawa, 2000 (in Polish)

---

# Preliminary Investigations Regarding the Blood Volume Estimation in Pneumatically Controlled Ventricular Assist Device by Pattern Recognition

D. Komorowski<sup>1,2</sup> and M. Gawlikowski<sup>2</sup>

<sup>1</sup> Institute of Electronics, Division of Microelectronics and Biotechnology, Silesian University of Technology, Gliwice, Poland  
dkomorowski@polsl.pl

<sup>2</sup> Artificial Heart Laboratory, Institute of the Heart Prostheses, Zabrze, Poland  
mgawlik@frk.pl

**Summary.** Mechanical heart supporting is one of possible therapies in case of circulatory system insufficiency. It is realized by application of various types of Ventricular Assist Devices (VAD). For monitoring, optimizing and automatic control of heart supporting process it is necessary to measure blood volume pumped through the VAD during single cycle. The paper presents a novel approach to the problem of reliable VAD's output estimation by the use of membrane's shape pattern recognition and related blood's chamber volume determination. Among many possibilities, the application of pattern recognition system has been decided leading to the promising results and making possible implementation of the automatic control algorithm in the future for the whole measurement process.

## 1 Introduction

Many diseases, like coronary and valve disease, myocarditis and cardiomyopathy may cause circulatory system insufficiency [1]. First sort of therapy is pharmacological treatment but in case of complete cardiac failure the heart transplantation is indispensable. About four thousand people are qualified to this treatment every year in Poland, but only 10% obtain organ to transplantation [2]. In case of acute or chronic heart insufficiency one of possible therapies is mechanical heart support with the use of special pumps VAD [3]. Because of diversified illness reasons, personal anatomical differences, various courses of disease and treatment methods it is necessary to adapt VAD operation to individual requirements of patient [3]. Moreover, the supporting conditions (like arterial systemic resistance) are not stable and it can change suddenly after medicine application or during normal, physiological cycle [4]. Monitoring, optimizing and control of heart supporting process require information about VAD output, defined as blood volume pumped through the device during its single operation cycle. At present, in extracorporeal VAD, the one method to realize noninvasive pump's

output measurement is ultrasonic flow meter utilize with sensor attached to canula [5]. Other examined methods did not give positive results [6] or they were hard to technical realization [7][8][9][10].

## 2 Methodology

Below described method deals with an idea of pattern recognition to overcome the difficulties related to the blood volume estimation in pneumatically controlled ventricular assist device (VAD). We have applied a classical method of pattern recognition to estimate a blood volume in VAD. Pneumatic ventricular assist device consists of the following components: polyurethane made blood and pneumatic chambers, disc valves seated on titanium rings and three-layers, flexible membrane, separated by graphite powder. Delivered through the connector pipe supplying air causes irregular membrane crimping and its collapsing or bulging (in both diastolic and systolic operation phase, respectively). The spherical shaped membrane is pliable and flexible, and it crimps irregularly during displacement between extreme positions. The membrane's shape irregularity is one of main obstacle during blood volume measurement process. Theoretical consideration revealed, that the basis of measurement process should be analysis of the whole solid, formed by blood or pneumatic chamber and membrane. It was explained in previous papers that such approach is proper and obtained results well correlate with reference method [7]. The technical realization difficulties of acoustics' waves utilization persuaded authors to exploit membrane imaging and its pattern analyzing. The experiments were performed on components of clinically used blood pump POLVAD. Examined membrane fixed to pneumatic chamber was mounted on stand (Fig. 1, 2).

The central point of membrane was marked by graphic indicator (1cm diameter circle). Image acquisition was realized by CMOS wide-angle lens camera, with resolution 640/480/24bit color depth. In order to acquire various types of membrane image, two camera arrangements were tested. In the first case lens was mounted 10cm from membrane base plane so as to register whole membrane image. In the second case lens mounting distance was decreased to 3cm so as to register maximal square of graphic indicator. The focal distance was constant and it was set so as to obtain an acuteness at membrane's basis plane. The polyurethane made membrane has transparent, reflected surface but graphite separating layer is black. To facilitate graphic indicator recognition it was filled red. System was lighted by one point diffused white light source of color temperature about 2570K. Membrane bulging and collapsing was caused by piston system via operating liquid. This kind of supplying allowed precise volume determination and simultaneously it was a reference. The fluid volume in VAD is proportional to location the pump membrane. We proposed to measure an area of small piece of pump membrane. The area of chosen region of membrane measurement allows to estimate the fluid volume. The proposed method bases on comparison of the average color in special coordinate region with last colors of analyzed image. The coordinate region is very easy to find because the location



Fig. 1. Laboratory stand, lens distance = 10 cm



Fig. 2. Laboratory stand, lens distance = 3 cm

of special region of membrane is known and depend on camera settings only. To validate the suggested method both suitable simulations and animals measurements (in the second stage) have been performed. First the lab stand including both VAD and pneumatically driven control unit has been designed with intention to acquire data characterizing behavior of the system under application. The detail algorithm of proposed method is described below. The recorded image of the membrane is converted from RGB type image into an  $L \times a \times b$  image type. The  $L \times a \times b$  is one of device-independent standard color spaces defined



**Fig. 3.** Image of membrane and some of coordinate region

by CIE, Commission Internationale de l'clairage (International Commission on Illumination) [12].

Next stage is to calculate the mean  $\bar{a}_c$  and  $\bar{b}_c$  values of  $\mathbf{a}_c$  and  $\mathbf{b}_c$  (1, 2), for extracted obligatory chosen small triangular area which is placed in examination area, this area will be called coordinate region. An example of chosen area is shown on Figure 3.

$$\bar{a}_c = \frac{1}{m_c} \sum_{j=1}^{m_c} \frac{1}{n_c} \sum_{i=1}^{n_c} a_{cij}, \tag{1}$$

$$\bar{b}_c = \frac{1}{m_c} \sum_{j=1}^{m_c} \frac{1}{n_c} \sum_{i=1}^{n_c} b_{cij}, \tag{2}$$

where  $m_c$  and  $n_c$  are size values of smallest rectangular area which included coordinate region. These values serve as color marker in  $\mathbf{a}_c \times \mathbf{b}_c$  space. The color marker has an  $\bar{a}_c$  and a  $\bar{b}_c$  value. We can classify each pixel in the membrane image by calculating the Euclidean distance between that pixel and color marker (3). Next the average value of Euclidian distance is calculated (4).

$$\mathbf{d} = \sqrt{(\mathbf{a} - \bar{a}_c)^2 + (\mathbf{b} - \bar{b}_c)^2}. \tag{3}$$

$$\bar{d} = \frac{1}{m} \sum_{j=1}^m \frac{1}{n} \sum_{i=1}^n d_{ij}, \tag{4}$$

where  $m$  and  $n$  are size values of image. If the value of calculated distance is smaller then average distance multiplied by coefficient then the pixel would be labeled as a color marker pixel. The value of coefficient is experimentally chosen

and set between 0.37 to 0.56. The obtained image is converted into binary image in a simple way. If calculated distance is smaller then threshold value  $t_v$  (5) then appropriate pixel is set as a '0' value and respectively if calculated distance is bigger or equal the pixel is set as a '1' value.

$$t_v = \leq 0.37, 0.56 \geq \times \bar{d}. \tag{5}$$

Next the image is improved by using an algorithm based on morphological reconstruction to fill holes in the binary image [11][12]. After that, all connected objects containing fewer than 50 pixels are removed from image of membrane. In the last step the area of obtained image is calculated by summation of those pixel which value is set to '1'. The result is proportional to the volume of ventricular assist device.

### 3 Results

The results obtained with the help of elaborated method show that the reliable blood flow estimation in the VAD is possible but requires additional researches and installation of CMOS wide-angle lens camera.

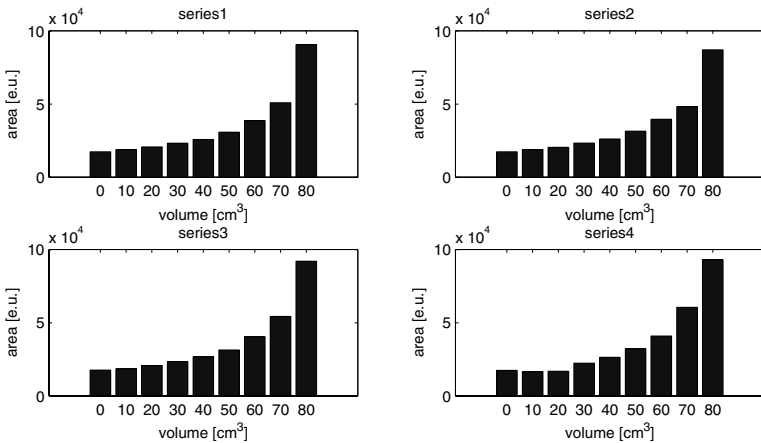


Fig. 4. An example results for distance lens = 3cm

One of the goal of this work refers to the elaboration of the method allowing reliable estimation of the VAD output blood flow. Figures 4 and 5 depict some results of measured area which have been obtained for different fluid volume of VAD chamber and for two lens distance. These results are presented as a area versus volume. On figures 6 and 7 the following steps of image processing are presented. First is a raw image of a membrane from camera, second presents the coordination region, third presents extracted red graphic indicator. Next depicts the result after binary process, small particles filling and filtration operation.

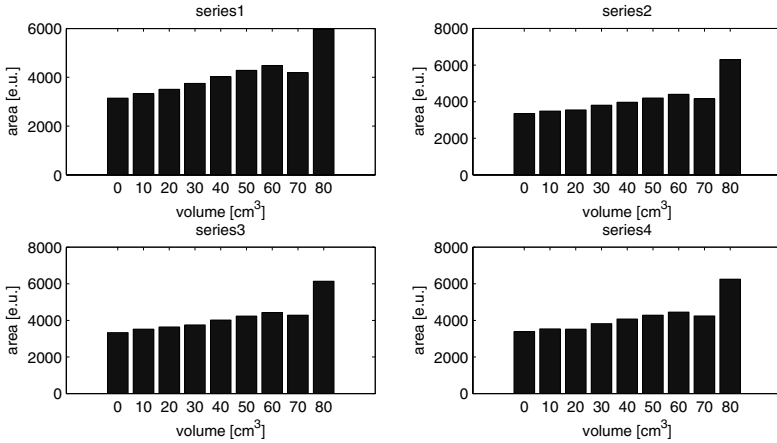


Fig. 5. An example results for distance lens = 10cm

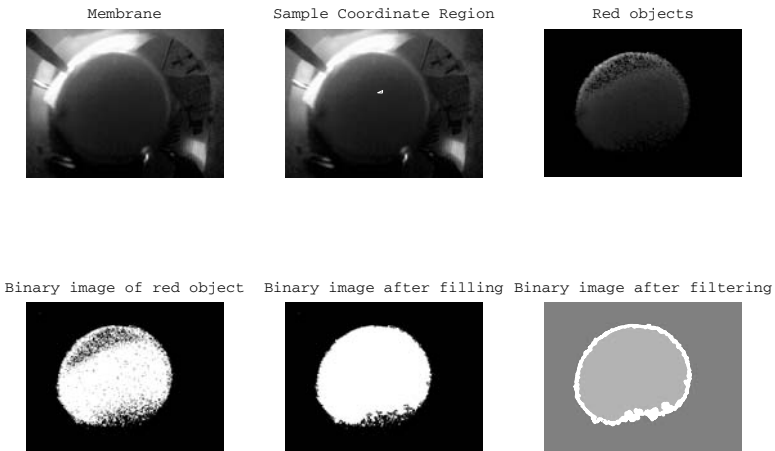


Fig. 6. An example results for distance lens = 3cm

## 4 Discussion

The main obstacle during pattern recognition and characteristic region's area calculating are artifacts formed by light reflection on membrane surface. Despite of dispersed light application and special anti-shadow shielding some of artifacts appeared and caused stochastic error. To limit this phenomenon in further experiments authors are considering the physically negation of characteristic region, namely covering whole membrane by nonreflective red film and forming a 1cm diameter non-covered indicator at the central point of the membrane. The presented approach concerning estimation of the output hydrodynamic flow going

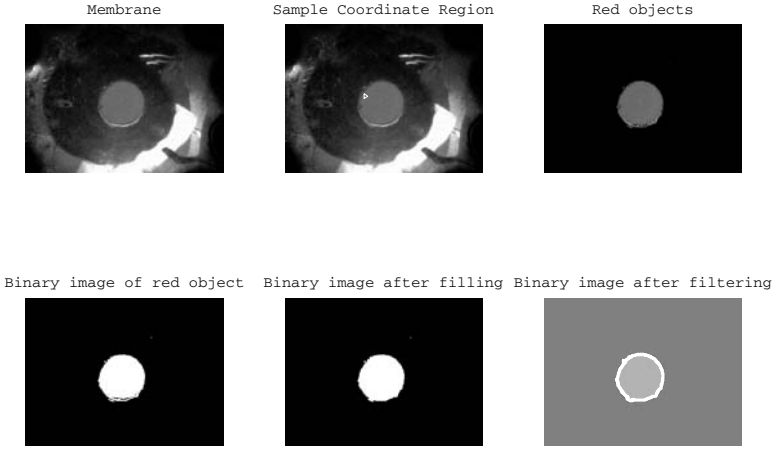


Fig. 7. An example results for distance lens = 3cm

out from VAD allows obtaining important physiological information required by doctors during the circulatory system assistance without additional risk to the patient.

## 5 Conclusions

Current research is concentrated on implementation of the elaborated method with the help of state of the art DSP's, available on the market and afterwards if that implementation is successful the whole procedure will be installed in the control unit responsible for driving VAD. Presented method is efficient, relatively simply to implementation and suitable to perform accurate measurement of blood chamber volume. Any modifications of present POLVAD construction are strongly undesirable, because of the subminiature CMOS camera equipped with optical lens will be applied in further investigations. The optical features of membrane's surface impose additional requirements on lighting. To avoid disturbing reflections the dispersed light is necessary. The specially prepared white LED may be utilized.

## References

1. Red Religa Z (1993) An outline of cardio-surgery. PZWL, Warsaw,(in Polish)
2. Internet: [www.sztuczneserce.pl/misja.html](http://www.sztuczneserce.pl/misja.html)
3. Religa Z, Kustosz R (2002) Mechanical heart supporting. Coronary vessels surgery. PZWL, Warsaw, str. 158-166. (in Polish)
4. Li J (2004) Dynamics of the vascular system. World Scientific Publishing. Singapore
5. Guyton A C (1991)Textbook of Medical Physiology. SAUNDERS Co., Philadelphia



6. Gawlikowski M, Pustelny T, Kustosz R (2006) Non invasive blood volume measurement in pneumatic ventricular assist device POLVAD. *Molecular and Quantum Acoustics* Vol. 27, p. 97-106
7. Gawlikowski M, et al (2006) Preliminary investigations regarding the possibility of acoustic resonant application for blood volume measurement in pneumatic ventricular assist device *Molecular and Quantum Acoustic*, Vol. 27, p. 89-96
8. Komorowski D, Tkacz E, Kustosz R (2002) An Application of the Neural Network for Output Flow Estimation in the Pneumatically Driven Polish Ventricular Assist Device (POLVAD) 29th ESAO Congress European Society for Artificial Organs, Vienna, Austria, *The International Journal of artificial Organs*, vol. 20, no. 10
9. Komorowski D, Tkacz E, (2005) Output Flow Estimation of Pneumatically Controlled Ventricular Assist Device with the Help of Artificial Neural Network', *Computer Recognition System*, Proceedings of the 4th International Conference on Computer Recognition Systems CORES'05, p. 561-568, ISSN 1434-9222, Springer-Verlag Berlin Haidelberg
10. Komorowski D, Pietraszek S, Darlak M (2006) Pressure and Output Flow Estimation of Pneumatically Controlled Ventricular Assist Device (VAD) with the help of Both Acceleration and Gyro Sensors, World Congress on Medical Physics and Biomedical Engineering, Seoul, Korea, Issue on CD
11. Soille P (1999), *Morphological Image Analysis:Principles and Applications*, Springer-Verlag, pp. 173-174
12. Matlab (2006), *Image Processing Toolbox Manual*

---

# 'Sonar' – Region of Interest Identification and Segmentation Method for Cytological Breast Cancer Images

Tomasz Nieczkowski and Andrzej Obuchowicz

Institute of Control and Computation Engineering, University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, Poland

T.Nieczkowski@weit.uz.zgora.pl, A.Obuchowicz@issi.uz.zgora.pl

**Summary.** In this paper we describe a novel image analysis method for the identification of region of interests and segmentation. The method was developed for the segmentation of the breast cancer cytological images, but it is also applicable to other image types. The primary technique of the method is a classification of the image pixels based on a spatial analysis of a feature variance in the pixel neighbourhood. The result of the analysis is association of a vector of values bounded to  $[0..1]$  to each pixel. Each of the vector values depends on a set of the feature differences between pairs of subsequent regions situated along symmetrical neighbourhood bearings. The vectors are then matched with a template vectors prepared for each of the types of searched regions of interest. The method allows for accurate localization of artefacts like edges, and subsequent segmentation of the image.

## 1 Introduction

Breast cancer is becoming the most common cancer disease among women over the world [1]. Automatic diagnostic systems of breast cancer have a long history and are still dynamically evolving [2]. One of the most common and the most efficient breast cancer diagnostic test is the Fine Needle Biopsy (FNB). In automated diagnostic systems based on the technique, a set of problems regarding automatic image processing needs to be solved.

The base for the diagnosis processed with the Fine Needle Biopsy is the evaluation of a quantitative morphological measure set of cellular nucleuses identified in the examined microscope specimen. To carry on the evaluation process automatically, efficient tools for accurate segmentation of the nucleuses must be provided.

Until now, many segmentation methods were proposed [3][4][5], but application of the methods to a specific, real-world, cytological image set introduces new problems. In this paper a method of region of interest identification and segmentation, based on a spatial analysis of a feature variance in the pixel neighbourhood is proposed. The result of the application of described algorithms is detection of the edges of nucleuses and the image segmentation. In the development and verification process of the method the authors used cytological images

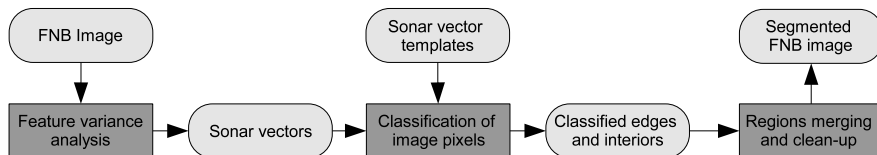
from the benchmark database ONKOMED [6] provided by the Department of Pathomorphology of the District Hospital in Zielona Góra, Poland.

The paper is organized as follows. Section 1 presents an introduction to the subject being considered. Section 2 contains a detailed description regarding the method. The obtained results are presented in Section 3. The last section concludes the paper.

## 2 Sonar Method

The segmentation process with the Sonar method consists of three stages. The first stage is the analysis of a feature variance in the pixel neighbourhood. This is the primary technique of the method. The name of the method originate from a similarity of this mechanism to the physical phenomenon of the sonar sound wave speed alternation between water regions of different physical condition (temperature, pressure, etc.).

The result of the first stage is a matrix of sonar vectors representing the variance of the examined feature in each pixel neighbourhood. The next stage is a classification of the image pixels based on a comparison of the sonar vectors with the primarily prepared template vectors. Classified pixels produce a set of regions, which are merged in the last stage. Figure 1 shows the scheme of the method. Details of the process stages are explained in the subsequent sections.



**Fig. 1.** Sonar segmentation method scheme. Dark grey blocks – process stages; light grey blocks – input and output objects.

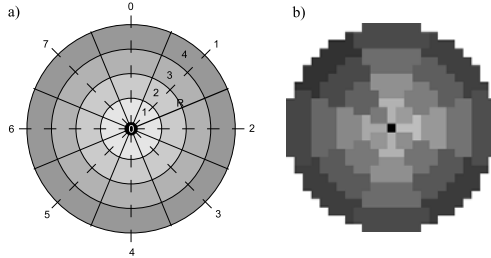
The neighbourhood for the feature variance analysis is defined as a set of concentrically situated rings of pixels. All of the rings have the same width and together they cover a circle, which centre is in the examined pixel. The circle is then partitioned into a number of slices along concentrically located bearings. Intersections of the rings and the slices are the neighbourhood sectors. The number of sectors in each ring is the same, and it equals to the number of the neighbourhood bearings.

Formally, the sonar neighbourhood  $N_{sonar}$  can be defined as quadruplet:

$$N_{sonar} = (R, r, s, \theta) \tag{1}$$

where  $R$  is the neighbourhood range,  $r$  is the number of rings,  $s$  is the number of bearings (sectors in each ring) and  $\theta$  is the angle between the zero-sector axis

and the 'north' bearing. The neighbourhood bearings are numbered from 0 and rings are numbered from 1. The zero-ring is the examined pixel itself. Figure 2 shows a visual model of the sonar neighbourhood for  $s = 8$ ,  $r = 4$  and  $\theta = 0$  (a) and its discrete realization for  $R = 12$  (b).



**Fig. 2.** Sonar neighbourhood model for  $s = 8$ ,  $r = 4$  and  $\theta = 0$  (a) and its discrete realization for range  $R = 8$  (b)

The sonar neighbourhood applied to an actual pixel allows for calculation of the sonar vector  $\sigma$ . The length of the vector equals to the number of bearings, so each of the vector values represents the variance of the examined feature along single bearing. The values bounded to  $[0..1]$  are calculated on the basis of a set of differences of the feature statistics between subsequent sectors along the bearing. The statistics used in the presented application of the method was arithmetic mean of the luma component of the YCbCr colour model.

For bearings with a very slight variance of the feature, the sonar vector values tend to 1 and decrease for greater variances of the feature. The value  $\sigma_s$  of the sonar vector  $\sigma$  along the  $s$  bearing is calculated as follows:

$$\sigma_s = \begin{cases} 1 - \Delta_s & \text{if } \Delta_s \leq 1; \\ 0 & \text{if } \Delta_s > 1. \end{cases} \tag{2}$$

$$\Delta_s = \sum_{i=0}^{r-1} \delta(|f_{s,i+1} - f_{s,i}|)w_i \tag{3}$$

where  $f_{s,i}$  is the feature statistics of the sector at  $s$ -bearing of the  $i$ -th ring, and  $w_i$  is the weight for the boundary between  $i$  and  $i + 1$  rings. The  $\delta$  function is a monotonous nondecreasing function, bounded to  $[0..1]$ . For the purpose of this research following function was used:

$$\delta(t) = \begin{cases} \frac{t}{d} & \text{if } t \leq d; \\ 1 & \text{if } t > d. \end{cases} \tag{4}$$

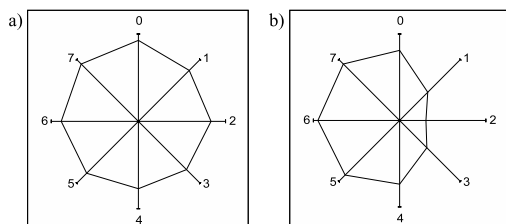
where  $d > 0$  is the sensitivity parameter defining the threshold value of the statistics difference, over which the value of  $\delta$  function remains 1.

The influence of the feature statistics difference between subsequent rings should be dependent on the distance of the ring from the examined pixel. The difference between rings located close to the neighbourhood centre should have greater impact on the sonar vector value than the difference between outer rings. To achieve this effect, the weight for the  $i$ -th and  $i + 1$ -th rings boundary could be defined as follows:

$$w_i = \frac{1}{2^{i_w}} \tag{5}$$

where  $w$  is the arbitrary weighting factor.

The calculated sonar vector can be visualised with a radar plot. A shape of the plot expresses the feature variance in the neighbourhood of the examined pixel. Later in this paper, the sonar vectors visualisations are stated as sonar views. Figure 3. shows sample sonar views for pixel located inside a uniform region (a) and for a pixel located near the right edge of the region (b).



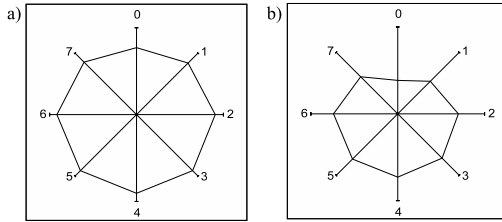
**Fig. 3.** Sample sonar views: (a) for a pixel inside of an uniform region, (b) for the pixel near the right edge of the region

The sonar vector matrix representing the image, prepared as described above, is used for a classification of the image pixels. The aim of the classification is to associate to each pixel one of  $k$ -classes labels. Each class represents a single artefact type which has to be identified for the image segmentation stage. In the application of the Sonar method to cytological images, two classes were used: the edge class and the interior of a uniform region class.

To perform the classification process, template sonar vectors are prepared for both classes. The preparation is carried out automatically, by the classification of another cytological image, performed with a binary, manually-segmented reference image from the benchmark database. The classification divides the image pixels into two sets: one for edges and the second for interiors of uniform regions. Classified sonar vectors for the cytological image are then calculated. Mean values of each class vectors along each bearing, produces the mean sonar vector, which is the template for the class.

A sonar view shape for the 'edge' class depends on the position of the edge. As shown in the Figure 3.(b), the sonar value for the bearing perpendicular to the edge is less than values for bearings inwards the uniform region. Thus, for

the 'edge' class, the number of sonar vector templates should be the same as the number of bearings. To reduce the calculation cost regarding introduction of a number of the edge class templates, a normalization of sonar vectors is performed beforehand. The normalization process of the sonar vector is a cyclic rotation of its values, until the lowest value is at the zero-bearing. After such an operation, each of the 'edge' class sonar vectors equals the edge on the 'north' bearing, so a single edge class template can be produced. Additionally, for the purpose of further stages of the method, the number of rotation steps performed in the normalization process for each sonar vector is saved. Later in this paper, the number of rotation steps is simply stated as 'rotation'. The result of the sonar vector templates preparation is shown in Figure 4.



**Fig. 4.** Sonar view templates for the 'edge' (a) and the 'interior' class (b)

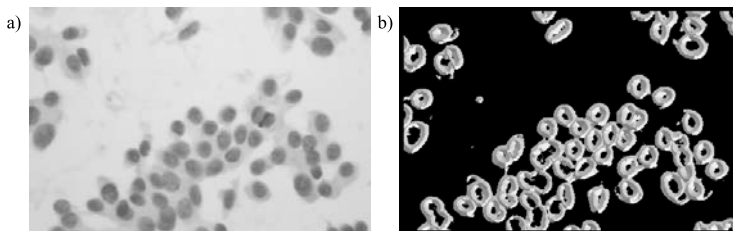
Actual classification of the image pixels is preformed by finding the minimal mean squared error between the pixel sonar vector and each of the template vectors. The mean squared error  $MSE$  of the sonar vector compared to the  $C$ -class template vector is defined as follows:

$$MSE_C(\sigma) = E((\sigma - \sigma_C)^2) = \frac{1}{s} \sum_{i=0}^s (\sigma_i - \sigma_{Ci})^2 \tag{6}$$

where  $\sigma$  is the sonar vector,  $\sigma_C$  is the sonar vector template of the class  $C$ , and  $s$  is the number of bearings. The pixel is labelled with the label of the class with the minimal mean squared error. Figure 5. shows a sample cytological image and the result of the pixel classification with Sonar. For the 'edge' class the saved number of the rotation steps is marked with the greyscale, from white colour for the bearing 0 to dark grey for the bearing 7.

The next step of the method is the actual segmentation of the image to the 'nucleus' and the 'background' labelled segments. The process described in previous sections associates a label of one of the two classes to each pixel. These labels, when combined with the rotation of the 'edge' class sonar vectors, produces a 9-label classification of the image pixels. Each consistent, separated group of pixels with the same label is considered a segment. To achieve the segmentation of nucleuses, each group of segments that forms a nucleus need to be merged.

As can be observed in the Figure 5, actual edges of objects are surrounded from both sides with the edge segments. One side is the nucleus boundary, and



**Fig. 5.** Cytological image (a) and the result of the pixel classification with Sonar (b)

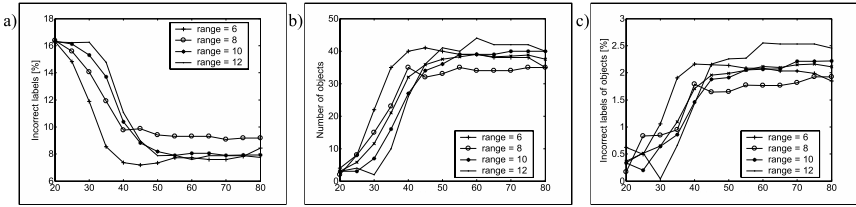
the other is the background boundary. Adjacent segments of the actual edge have opposite rotation, so it can be distinguished which neighbouring interior segment should be merged with the edge segment. Along a single boundary, the edge segments should be merged if the distance between their rotations equals 1. The segments should be also merged with the interior segment located below the edge. The result of the segment merging process is the segmentation of the image along actual edges of objects.

### 3 Results

To assess the quality of the Sonar segmentation method, the segmentation of a control image for a range of sonar neighbourhood parameters was performed. The result of the segmentation for each set of parameters was compared to the regarding manually-prepared reference segmentation image, obtained from the benchmark database. Three quantitative measures were used for the segmentation assessment [7]: the number of objects found, the proportion of incorrectly labelled pixels in the whole image and the same proportion limited to pixels of identified objects. Figure 6 shows the results of the assessment for a number of sonar neighbourhood ranges and sensitivity parameter values, with fixed numbers of 8 bearings and 4 rings.

As shown in the figure above, for the sensitivity values greater than 50, all three measures remain stable. For these values over 40% of objects were found. The mean proportion of incorrectly labelled pixels was at the level of 8.1%, which is a relatively high number. The cause was, that only 38 out of 74 of the nucleuses were identified on average. However, in the case of the cytological images segmentation, the accuracy of the segmentation of identified objects is more important than the total number of the objects found. The accuracy can be expressed with the proportion of incorrectly classified pixels, limited to the identified objects. In the presented results, the proportion was at the level of 2.1%. This measure, along with a visual examination of the segmentation result, allows ranking the Sonar segmentation applied to cytological images moderately high.

To assess the method robustness, the authors also assessed the segmentation results of the same image with a Gaussian blur and uniform noise added. The



**Fig. 6.** The assessment of the control image segmentation: the proportion of incorrectly labelled pixels in the whole image (a), the number of identified objects (b), the proportion of incorrectly labelled pixels limited to identified objects (c)

proportion of incorrectly labelled pixels in the whole image was 7.0% and 8.1% respectively. The results obtained from the deformed images segmentation shows that the Sonar segmentation method performs well on blurred and noised images. Actually, the result for the blurred image was slightly better than for the original image. This feature of Sonar method is very important for the segmentation of a set of images with common deformation, for example images acquired with low-quality optical equipment.

### 4 Conclusion

In this paper, a novel region of interest identification and segmentation method is proposed. The method bases on the analysis of the image pixel feature variance in a wide neighbourhood of the examined pixel. The Sonar method, can be applied to the localization of a number of the image visual artefact types. With the additional segment merging stage, the method can be also applied to the image segmentation. In the presented application to the cytological images segmentation, edges and uniform regions were detected.

Due to the necessary calculation of a large number of statistics for pixels neighbourhoods sectors, the method has moderately high computation cost. The number of stages, which need to be performed to obtain the segmentation of the image, increases the complexity of the technique and makes it more difficult to implement. The counterbalance for these disadvantages is there are no dependences among the image pixels states at the sonar vectors computation stage, so it can be easily parallelized. To reduce high computation cost of the method, the sonar vector matrix obtained from the first stage can be treated as the image model, and utilized in multiple processing schemes on further stages. For example, the higher number of sonar vector templates can be introduced and a complex analysis of class membership probabilities can be performed at the classification stage.

The Sonar method is adaptable and extensible. It can be easily adjusted to actual application, by setting a number of parameters, such as the neighbourhood range and shape, the number of rings and bearings or the sensitivity and weighting factor. Additionally, an alternative sonar vectors calculation function



or a classification error function can be used. In the presented application, the sonar vector values were calculated on the basis of mean luma changes between subsequent rings of the neighbourhood. It is also possible to use another pixel feature, or even a set of combined features.

The extensibility and adaptability of the Sonar segmentation method allow treating it as a general framework, which can be applied to analysis of any type of images containing repeatable visual structures. In the further research the authors plan to adjust the method to achieve better results of segmentation for a wider set of images from the benchmark database. The ultimate goal of the research is to provide a set of techniques and algorithms based on the Sonar framework for an actual FNB breast cancer automated diagnostic system.

## Acknowledgement

This work has been partially supported by the Integrated Regional Operational Programme (Measure 2.6: Regional innovation strategies and the transfer of knowledge) co-financed from the European Social Fund.

## References

1. Marciniak A, Monczak R, Kołodziński M, Prętki P, Obuchowicz A (2004) Fine Needle Biopsy Breast Cancer Diagnostic Benchmark (in Polish). Sztuczna inteligencja w inżynierii biomedycznej - SIIB 2004, Wyd. PreTEXT, Kraków
2. Kimmel M, Lachowicz M, Świerniak A (Eds.) (2003) Cancer Growth and Progression, Mathematical Problems and Computer Simulations, Int. J. of Appl. Math. and Comput. Sci., Vol. 13, No. 3, Special Issue
3. Gonzalez R, Woods R (2002) Digital Image Processing, Prentice Hall
4. Pratt W (2001) Digital Image Processing, John Wiley & Sons
5. Russ J (1999) The Image Processing Handbook, CRC Press
6. Marciniak A, Obuchowicz A, Monczak R, Kołodziński M (2005) Cytomorphometry of Fine Needle Biopsy Material from the Breast Cancer, Proc. of the 4th Int. Conf. on Comp. Recogn. Systems CORES'05, Adv. in Soft Computing, pp. 603-609, Springer
7. Zhang Y J (1996) A survey on evaluation methods for image segmentation. Pattern Recognition, Vol. 29. No. 8, 1335-1346, Elsevier Science Ltd

---

# Validation of Automatic ECG Processing Management in Adaptive Distributed Surveillance System

Piotr Augustyniak

AGH University of Science and Technology, 30 Mickiewicza Ave. 30-059 Krakow  
Poland  
august@agh.edu.pl

**Summary.** This paper presents the problem of diagnostic parameters quality control in the adaptive ECG interpretation system. The quality estimation is used for seamless adaptation of processing being a part of the optimization feedback. Therefore, quality control procedures - unlike in conventional systems - are embedded in the software. The proposed methodology is based on two principal concepts: diagnostic results priority depending on the patient status and the data convergence used as description of the adaptive system response to a transient in patient status. The paper includes an example of testing a surveillance network prototype with use of proposed parameters, demonstrating their high usability.

## 1 Introduction

Automated distributed diagnostic systems due to their wide spread and direct impact to the life quality are currently considered as very interesting and fast growing branch of telemedicine. Wearable recorders of today, although using the latest technology, functionally follow the bedside interpretive electrocardiograph [4] [3] concept formulated over 30 years ago [5]. Without the cooperative adaptation of the elements, surveillance networks are rather simulating a group of independent cardiologists using rigid interpretation procedures. Our approach, proposed in a series of papers [1] [9] [2] as the alternative, yields unprecedented personalization or diagnosis-oriented processing and thus better simulates the seamless presence of a cardiologist. The remote recorder, thanks to the reprogrammability, continuously adapts the ECG signal interpretation process to several prioritized criteria of medical and technical nature. The process is designed as distributed and performed partially by separated thread on the supervising server (network node) and partially by the agile software of the remote recorder. Important novelty is also the use of digital wireless link in a bi-directional mode for patient and device status reporting but also for management and control of the remote software, requests for adaptation of report contents and data priority and reloading of software libraries as necessary.

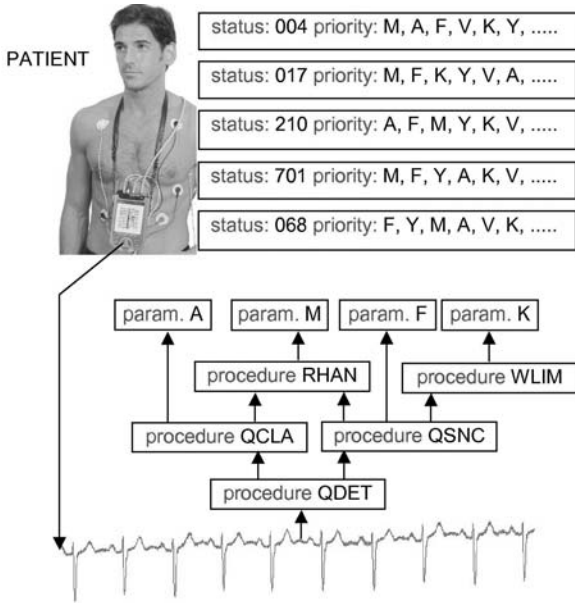
Such adaptive systems are expected to create new opportunities for the automated ECG processing, making the interpretation closer to the human way of reasoning and using some experimentally derived rules of cardiologists behavior. Until today they bring rather scientific challenges, revealing new unexploited areas present in clinical practice but not covered by the standards, recommendations or guidelines [8]. Comparing to conventional solutions, adaptive systems are much more complex and, in particular their correct behavior is not defined in a unique way.

This paper revises the notion of diagnostic data quality and quality monitoring in adaptive automated ECG interpretation systems. Comparing to the rigid interpretation systems, the quality is not only a performance factor [6], but also an important continuous measure of system compliance and error signal for the feedback modifying the interpretation software. Therefore seamless estimation of diagnostic data quality is an integral part of the system and determines its behavior. The proposed quality assessment system is based on the idea of prioritized convergence presented in this paper together with first results of tests performed on a developed prototype software using a modifiable subroutines in chain of distributed ECG interpretation process.

## 2 Materials and Methods

### 2.1 Overview of Automatic Software Management

Since the optimal patient description depends on his status, variable priority factors are attributed to particular medical data and consequently to the branches of the interpretation process (fig. 1). Data priority implies also variable expectance about the accuracy and reliability of particular parameters. Unfortunately, calculation of all data with maximum precision may not be technically feasible in a battery-operated wearable recorder, moreover transmission of non-relevant data rises the costs of wireless communication channel. The proposed software management uses sorted repositories of procedures for principal ECG interpretation steps. Each task-oriented repository contains functions of various algorithms (i.e. originating from various inventors), different complexity, resources requirements and performance, but easy to substitute each other thanks to a standardized interface. The repository sorting is based on the correlation of the computation complexity and the result accuracy, commonly observed in software engineering. The management of interpretation process considers the patient-dependent requirements, current quality estimate and technical conditions, and continuously seeks for best available accuracy to complexity ratio yielding the expected quality of diagnostic result (fig. 2). The processing setup is revised each time the accuracy expectations are not met or with variations of patient status (i.e. changes of diagnostic goal) and technical environment changes (e.g. transmission conditions). The feedback configuration data is sent



**Fig. 1.** Patient status-dependent priority of results from particular processing branches of ECG interpretation

via wireless channel to the remote recorder in order to replace and re-link particular procedure or procedures chain by their upgraded or downgraded alternative versions.

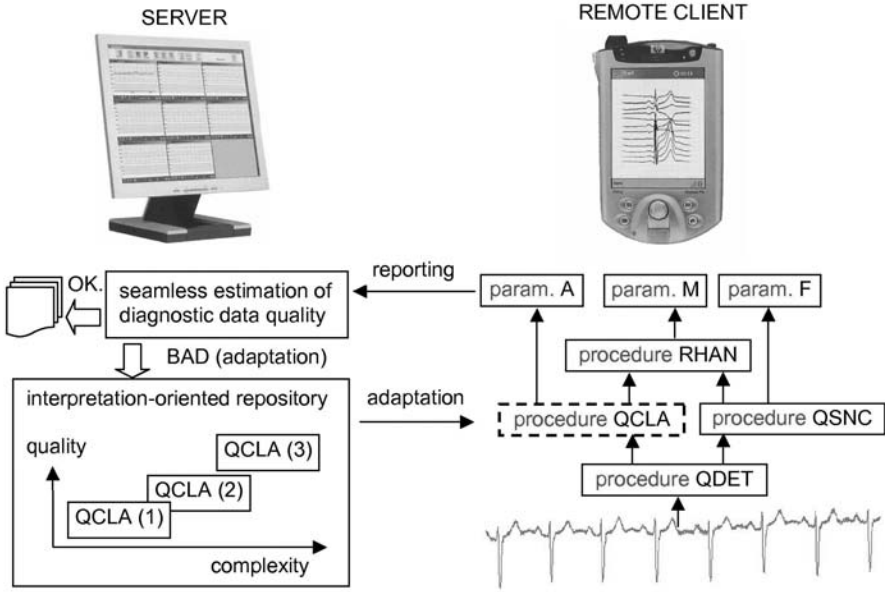
**2.2 The Data Convergence for Estimation the Quality**

Random redundant signal re-interpretation is proposed as a tool for occasional monitoring of remote interpretation reliability. This procedure uses bi-directional transmission and begins with the raw signal request issued by the supervising server. The remote recorder does the interpretation independently and besides the diagnostic result returns the raw electrocardiogram as well. The raw signal strip is interpreted by the server software in conditions of unlimited resources availability and thus the results are used as reference in comparison to those received from the remote interpretation. Any difference  $\delta_i$  is statistically processed and compared to the tolerance thresholds  $\epsilon_i$  with consideration of current data priority  $w_i$  (1).

$$Q = \sum_i |\delta_i| \cdot w_i \tag{1}$$

where:  $\delta_i = d_{i,k} - d_{i,ref}$ ,  $d$  is the diagnostic value of the parameter  $i$  and  $k$  is the modification attempt number.

Every outstanding value is a background for modification of the remote recorder interpretation software. Since the raw ECG is buffered in the remote



**Fig. 2.** The seamless estimation of diagnostic data quality in the remote recorder software adaptation feedback

recorder, the suspicious signal strip is interpreted again and the results are sent to the server are compared to the reference. The convergence represents the correctness of decisions made by the management procedure about the remote interpretation procedures. The software adaptation plays the role of a discrete feedback correcting the automatically made diagnoses and consequently the convergence of diagnostic results  $C$  is required for the stability of the adaptive interpretation system.

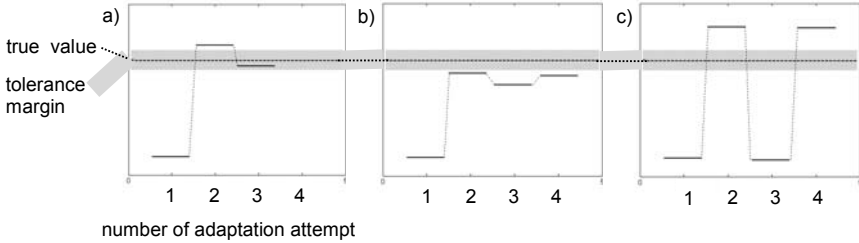
$$C = \lim_{k \rightarrow \infty} |d_{i,k} - d_{i,ref}| \tag{2}$$

If the software modification decisions are correct, the outcome altered by few attempts of interpreting software modification  $k$  approaches to the true value  $d_{i,ref}$ . The modification request request  $R$  is removed as soon as the outcome value  $d_{i,k}$  falls into a given tolerance margin  $\epsilon_i$  around its true value  $d_{i,ref}$ , and  $|d_{i,\infty} - d_{i,ref}| < \epsilon_i$  indicates that the system is stable.

Incorrect decisions lead to the growth of diagnostic outcome error  $\delta_i$  and imply even stronger request for modification  $R$ . The outcome value  $d_{i,k}$  may stabilize on an incorrect value  $d_{i,\infty}$ , where  $|d_{i,\infty} - d_{i,ref}| > \epsilon_i$  or swing the measurement range in response to subsequent trials (3).

$$\lim_{k \rightarrow \infty} |d_{i,k+1} - d_{i,k}| > 0 \tag{3}$$

In such case the system is unstable and the diagnostic outcome  $i$  does not converge to the true value  $d_{i,ref}$ (fig. 3).



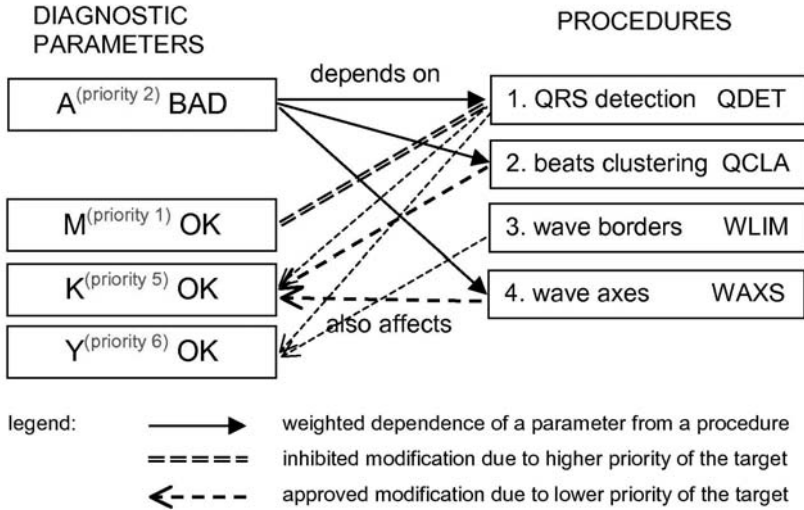
**Fig. 3.** Examples of possible system adaptation a) converging and stable, adaptation ends in 2 iterations; b) stable but not converging - the data error still results in adaptation request; c) not stable and not converging

### 2.3 Data Priority-Dependent Software Management

The overall estimate of diagnostic quality has to follow the variability of patient status and the data relevance. A straightforward approach assuming that quality estimates directly correspond to modified procedures is here impractical. The dependence between diagnostic parameters and interpreting procedures is much more complex (fig. 4). Each final outcome is calculated by a chain of several procedures affecting its quality. Conversely, each procedure, particularly at early processing stages, affects several diagnostic parameters. These relations, implied by the software architecture were compiled to a static list appended to each repository attributes and considered by the software management subroutine. In case the modification request  $R$  is issued by the particular parameter-oriented quality estimator, the list is hierarchically scanned in order to detect any conflict of interest between simultaneously affected data. This hierarchy depends on current patient status and approves the modification only in case of procedures contributing to parameters of lower priority.

## 3 Examples of Use and Results

The behavior of single client-server pair being a limited-scale ECG monitoring network prototype with auto-adaptive software was investigated with use of the proposed tools. The remote recorder was based on a PDA-class handheld computer with Bluetooth-connected ECG acquisition module (Aspekt 500 by Aspel, 8 channels, 12 bits, 500 sps) and bi-directional GPRS connection. The stationary server was a PC-class desktop computer with a static IP address and 100Mb Internet access running Linux-Apache-MySQL-PHP (LAMP) package. The database contained 857 signals composed of physiological ECG and 14 most frequent pathology representations joined artificially [7] [10]. The standard goal of such test is the assessment of the software adaptation correctness, but in our case main attention was paid to the usability of newly proposed quality estimation tools. Triggering error threshold for remote software modification was set accordingly to the diagnosis priority in four categories: 2% for QRS detection



**Fig. 4.** Example of dependence between diagnostic parameters and interpreting procedures influencing other outcomes of various priority

and heart rate, 5% for wave limits detection and ST-segment, 10% for morphology classification and 20% for other parameters. The overall distance  $Q$  in the diagnostic parameters hyperspace is expressed by the values of diagnostic parameters errors weighted by the priority of particular data (1). The correctness of software modification is expressed in technical aspect by the percentage of incorrect adaptation attempts (89 cases, 10,4%). As such were considered resources overestimation, leading to allocation violation (27 cases, 3,1%), and underestimation, resulting in suspending of the software upgrade when the upgrade was feasible (62 cases, 7,3%). In medical aspect, the correctness of interpretive software modification is expressed by the percentage of adaptation attempts leading to diagnostic parameters converging to the reference values (740 cases, 96,4%). In 28 cases (3,6%), however, diagnostic results quality was inferior in result of software modification.

**Table 1.** Cumulative percentage of remote diagnostic results convergence test after 1-4 consecutive steps of interpretation software modification

step no.	converging	non-converging
1	63,1	36,9
2	74,5	25,5
3	79,1	20,9
4	80,7	19,3

## 4 Conclusions

Presented method offers estimation of various new features emerging due to the ECG processing adaptivity. Two principal concepts presented in the paper (data convergence, priority-dependent software management) reveal high complexity of the auto-adaptive software as well as areas not fully covered by current medical recommendations. Principal elements of proposed quality estimation method was used for assessment of a prototype cardiac monitoring network. In this application our method contributed to final adjustment of the system properties in particular automatic decision making about further processing and reporting in a remote recorder.

## Acknowledgment

Scientific work supported by the Polish State Committee for Scientific Research resources in years 2004-2007 as a research project No. 3 T11E 00127.

## References

1. Augustyniak P (2005) Adaptive Wearable Vital Signs Monitor for Home Care and Sports. In: Kurzynski M, Puchala E, Wozniak M., Zolnierok A (eds) Computer Recognition Systems Proc. of the 4-th CORES'2005, pp. 469-476, Springer Verlag.
2. Augustyniak P (2006) The use of selected diagnostic parameters as a feedback modifying the ECG interpretation. *Computers in Cardiology*; 33:825-828,
3. Banitsas KA, Georgiadis P, Tachakra S, Cavouras D (2004) Using handheld devices for real-time wireless Teleconsultation. Proc. 26th Annual International Conference of the IEEE EMBS, pp. 3105-3108
4. Gouaux F, et al. (2002) Ambient Intelligence and Pervasive Systems for the Monitoring of Citizens at Cardiac Risk: New Solutions from the EPI-MEDICS Project *Computers in Cardiology* 29:289-292.
5. (1974) IBM Electrocardiogram Analysis Program Physician's Guide (5736-H15) 2-nd edition.
6. (2001) IEC 60601-2-47 Medical electrical equipment: Particular requirements for the safety, including essential performance, of ambulatory electrocardiographic systems.
7. Moody G (1993) MIT/BIH Arrhythmia Database Distribution. Massachusetts Institute of Technology, Division of Health Science and Technology, Cambridge, MA,
8. Straszeka E, Straszeka J (2004) Uncertainty and imprecision representation in medical diagnostic rules. *IFMBE Proc, Medicon 2004*, paper 172
9. Tadeusiewicz R, Augustyniak P (2005) Information Flow and Data Reduction in the ECG Interpretation Process In: Zhang YT, Xu LX, Roux C, Zhuang TG, Tamura T, Galiana HL (eds) Innovation from Biomolecules to Biosystems. IEEE 27-th Annual IEEE-EMBS Conference, paper nr 88.
10. Willems JL (1990) Common Standards for Quantitative Electrocardiography 10-th CSE Progress Report. Leuven: ACCO publ.,



---

# Estimation of a Muscle Force from a Mechanomyographic Signal During a Contraction of a Single Motor Unit

Piotr Kaczmarek and Andrzej Kasiński

Poznań University of Technology, Institute of Control and Information Engineering, 60-965 Poznań, ul. Piotrowo 3a  
Piotr.Kaczmarek@cie.put.poznan.pl

**Summary.** In this paper an estimation method of a muscle force generated during a single Motor Unit (MU) contraction from a mechanomyographic signal (MMG) is described. The MMG signal has been recorded with a microphone immersed in a paraffin oil, placed above the medial gastrocnemius muscle of a rat. The estimation process evaluates the muscle surface displacement and thus indirectly the force. In estimation procedure the oil medium model have been used. This model describes the propagation of the acoustic wave generated by the muscle surface displacement and recorded with the microphone. The estimation procedure uses the model as a reference model enabling estimation of the muscle surface acceleration being the model input, from the MMG signal which is the model output. The method have been applied to the signals recorded *in vivo*. The results obtained with the method reveal a good correspondence between the estimated and the directly measured force signals.

## 1 Introduction

The contraction force measurement is a one of the crucial issues in diagnostics of myopathies, sport medicine, rehabilitation as well as in functional electrical stimulation (FES) systems. The direct measurement of the force is difficult. For most of muscles, it requires using special exoskeletons with force or torque transducers. The exoskeleton may restrict the freedom of motion and changes the system dynamics. Moreover, there is a number of muscles (ie. spinal muscles) for which the contraction force cannot be directly measured due to apparently closed kinematic chains or poor accessibility.

There are few approaches enabling indirect force measurement, where the muscle contraction force is estimated from physiological signals generated during a contraction. The most known are the electrophysiological techniques such as the electromyography (EMG) and electroneurography (ENG). Measured signals are the central neural system (CNS) commands evoking the contraction. The muscle force is dependent not only on the CNS excitation, but also on an actual muscle state (ie. a fatigue, a muscle length etc.). As, the muscle state cannot

be easily estimated from the EMG or the ENG signals, this makes the force estimation hard.

Beside the electrophysiological signals there are also a mechanomyographic signals (MMG) generated by the muscle contraction. They are recorded by using laser distance sensors[6], accelerometers[8], or microphones [2, 1] located over the muscle belly. Most of authors agree that the MMG is due to a transverse movement of the muscle surface occurring during the contraction. The mechanism of the muscle surface movement is complex. Three major factors should be mentioned:

1. Muscle pennation. The pennate muscle has the muscle fibers fixed to the muscle surface at a non-zero angle. The muscle fibers innervated by the same motoneuron (Mn) create a motor unit (MU). The MU is the smallest muscle element which contract during activation of a single Mn. It is also known, that MU occupies a restricted area of the muscle volume. Thus, during a single MU contraction local, transverse surface deformations should be observed as it is shown in fig. 1. This issue have been explored and modelled by our group previously [3].
2. Muscle rotation. Assuming a muscle shape elasticity during a single MU contraction, the muscle can be assumed to be a rigid body subjected to a couple of forces ( ${}^zQ, -{}^zQ$ ). As a result of the MU contraction, the muscle rotation occurs.
3. Muscle diameter increase results from the muscle shortening and its hyper-elastic properties.

Using the membrane muscle model mentioned above, in which the MMG phenomenon was simplified, the muscle contraction force was considered to be a

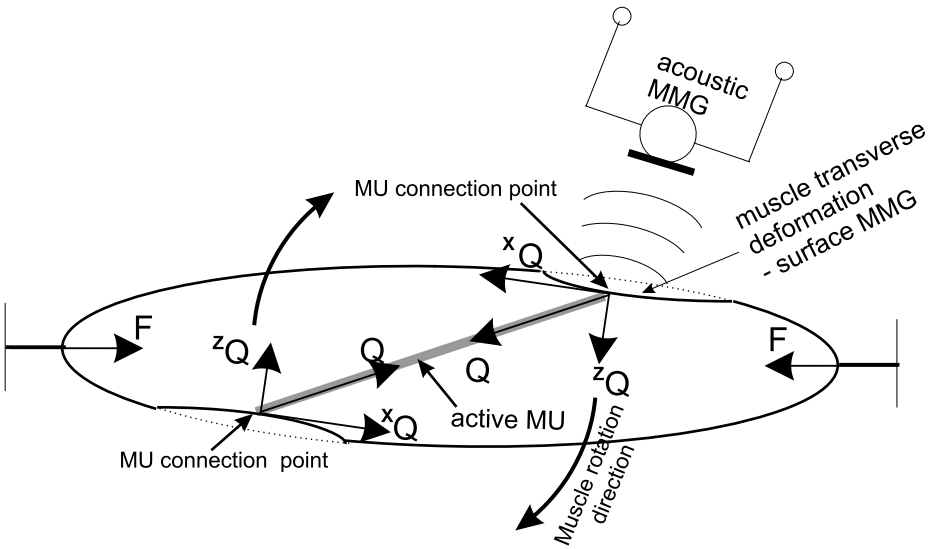


Fig. 1. The idea of the MMG signal generation

longitudinal component of the MU force ( ${}^xQ$ ) while the deformation was generated by the transverse component ( ${}^zQ$ ) as it was shown in fig. 1. This suggests, that the MMG might be a better tool for the estimation of the muscle force than the EMG and ENG. So far the investigation of the MMG signal has been focused on the relationship between the MMG and the muscle force, the fatigue level and the muscle length. Experiments revealed only some properties of the MMG such as a modulation of its frequency and amplitude with the force or the fatigue level changes. There are few models explaining the observed phenomena, although there is a lack of a model or a method for the estimation of the contraction force from the MMG signal.

Because many effects are generated during a voluntary contraction, it is proper first to develop the estimation method of the force during a single MU evoked contraction and then to extend it to the case of the voluntary contraction.

## 2 The Model of the MMG Signal Generation

Three major factors of the muscle surface displacement have been mentioned above. For a weak contraction the impact of the muscle diameter changes on a total muscle surface displacement can be neglected, but the observations suggest, that the muscle rotation and the local surface deformation should be taken into account. The local surface deformation is assumed here to be a linear and it has been described in detail in [3]. For the purpose of the present paper, it can be reduced to a static relationship. Thus, it can be approximated with the following equation:

$$z_l = k_l \cdot F \cdot \tan(\alpha_{penn}), \tag{1}$$

where  $z_l$  is a local transverse deformation of the muscle surface in a point of the MU fixture,  $F$  is a contraction force,  $\alpha_{penn}$  is the muscle pennation angle, and  $k_l$  denotes an elasticity coefficient for a local deformation.

The rotation of the muscle generated by the couple of forces, acting against the muscle elasticity and viscosity reaches a steady state. Assuming a simple dynamic model, this effect can be approximated with the following equation:

$$F = k_\alpha \alpha - c_\alpha \frac{d}{dt} \alpha, \tag{2}$$

where  $\alpha$  is a rotation angle,  $F$  is a contraction force and  $k_\alpha$  and  $c_\alpha$  are angular elasticity and dumping coefficients, respectively.

For the isometric contraction the rotation angle is small, thus the surface displacement due to the muscle rotation can be assumed to be proportional to  $\alpha$ . Assuming the muscle surface displacement to be a sum of the local muscle displacement (eq. 1) and the rotation displacement (eq. 2), the relationship between the contraction force and the total displacement can be approximated by the following equation:

$$F \approx k'z - c' \frac{d}{dt} z, \tag{3}$$

where the  $k'$  and  $c'$  are the resultant elasticity and dumping coefficients and  $z$  is the total muscle surface displacement.

The muscle surface acceleration generates the acoustic wave recorded as the MMG signal. The model of an acoustic wave propagation in the oil medium has been described in detail in [3]. The medium model was based on a Stiff Finite Element approach and consists of six cells elastic in the wave propagation direction and having viscotic properties in the transverse direction. In the model, the medium and the muscle surface interfere in the cell no 0, while the microphone is located in the cell no 1. The input to this model is the muscle surface acceleration acting at the cell 0 and the output is the pressure signal at the compartment 1 captured by the piezotransducer. The pressure is assumed to be proportional to the state variable describing the local displacement of the medium cell.

Our goal is to estimate the muscle surface acceleration only from the MMG signal recorded by a single piezotransducer, and then by applying a double integration and the relationship (3) to estimate the contraction force. This imply use of an inverse medium model, where the input is the MMG signal and the output is the muscle surface acceleration. However it is impossible to obtain because the direct model is not observable as the output signal contains only a single state space variable. Moreover, in a lumped model a wave propagation delay occurs, so the inverse model could be non casual.

### 3 Estimation Method

As the medium model is not invertible, another estimation procedure must be applied. In our approach the medium model is used as a reference model, and the muscle surface acceleration ( $\hat{U}$ ) is iteratively estimated by the minimization of the output prediction error ( $E$ ). It is defined as a difference between the measured MMG signal ( $Y$ ) and the predicted output ( $\hat{Y}$ ) obtained from the reference model. The Least-Squares Method (LSM) with en exponential forgetting [7] have been adopted to estimate the muscle surface acceleration signal. Contrary to the standard LSM, where the system parameters are estimated, here the medium model parameters are assumed to be known and the input signal is estimated.

In a general case the estimated system can be described by the following equation:

$$\hat{Y} = W \cdot \hat{U}, \tag{4}$$

where in the task of the input signal ( $\hat{U}$ ) estimation the vector  $\hat{Y}$  is an estimated output signal (MMG) and the matrix  $W$  denotes a deconvolution matrix defined in the following form:

$$W = \begin{bmatrix} h(0) & h(1) & h(2) & h(3) & \dots \\ 0 & h(0) & h(1) & h(2) & \dots \\ 0 & 0 & h(0) & h(1) & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}, \tag{5}$$

where  $h(\cdot)$  denotes the discrete impulse response of the reference medium model.

The parameter update law has the form:

$$E(n) = Y - \hat{Y}(n), \tag{6}$$

$$\hat{a}(n) = -P(n) \cdot W^T \cdot E(n), \tag{7}$$

$$\hat{U}(n+1) = \hat{U}(n) + \hat{a}(n), \tag{8}$$

where the estimator gain matrix  $P(n)$  is updated according the following rule:

$$\dot{P}(n) = \lambda P(n) - P(n)W^TWP(n), \tag{9}$$

$$P(n+1) = P(n) + k \cdot \dot{P}(n), \tag{10}$$

$\lambda$  being a constant forgetting factor.

## 4 Force Estimation from Signals Recorded *in Vivo*

### *Pre-estimation processing*

The MMG was recorded by a piezotransducer. Due to the piezotransducer impedance, drift in the recorded signal was observed. Moreover the MMG signal offset appeared. Therefore, before the estimation process, the linear regression coefficients are computed, in order to subtract the drift[4] from the MMG signal. The results of the pre-processing stage are shown in fig. 2A-C.

### *Estimation stage*

The input signal  $\hat{U}$  denoting the muscle surface acceleration, is estimated in a moving time-window enabling to apply the method on-line. The estimation window is shifted by 20% of its width, thus such an approach is similar to a parameters estimation method of a nonstationary process. The estimation procedure has following steps:

1. Initialization.
2. Iterative estimation of the input signal with a constant position of the window. The gain matrix  $P$  updated according eq. 9 and 10, with the forgetting factor  $\lambda = 0$ .
3. The time window is shifted by 20% of its width. The estimated input signal outside the window is not accounted for in the estimation procedure, however it is used to determine the location on time axis of the reference model response. The  $P$  is updated like in a nonstationary process estimation scheme assuming  $\lambda > 0$ . Moreover, the initial values of  $\hat{U}$  are taken from the previous estimation step, and the missing estimation values at the end of the vector  $\hat{U}$  are set to zero.
4. For the current window position, after updating  $\hat{U}$ ,  $P$ ,  $Y$  the procedure is continued by going to the step 2.

*Post-processing stage*

The estimated force is evaluated from eq. 3 where the muscle surface position is obtained by a double integration of  $\hat{U}$ .

$$\hat{F} = k' \iint \hat{U} - c' \int \hat{U}. \quad (11)$$

## 5 Results

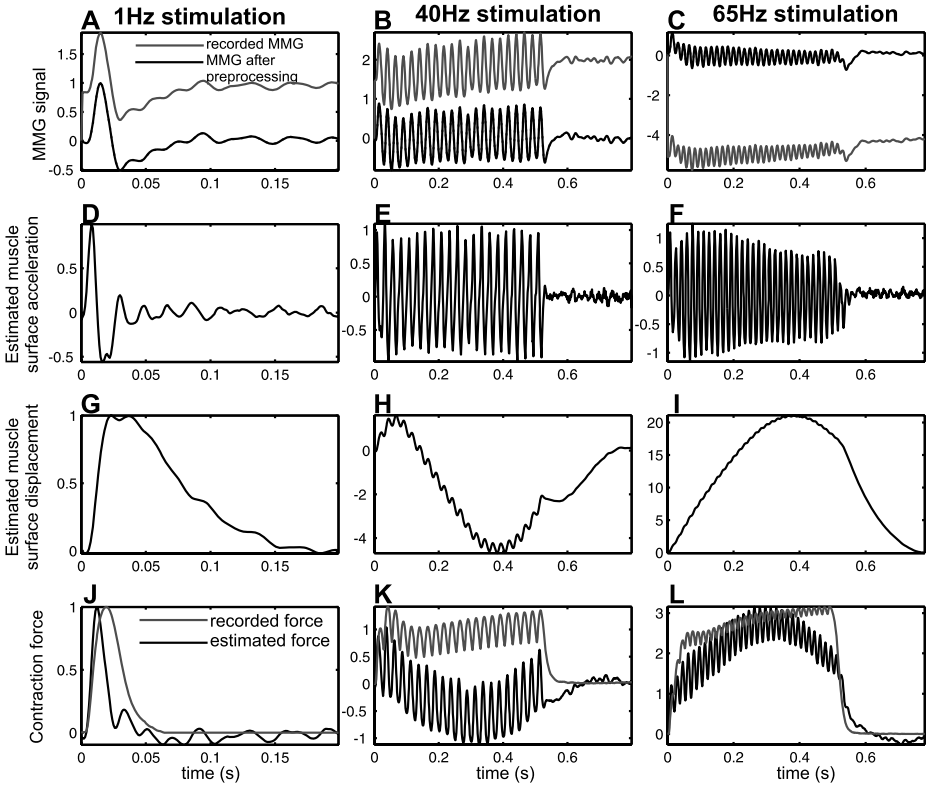
The estimation procedure has been tested on the MMG signals recorded *in vivo* from the medial gastrocnemius muscle of a rat. The muscle was stimulated with various frequencies and the force and the MMG signals have been recorded. The signals recordings was 300-800ms length and the time window was assumed to be 100ms. Moreover the step 2 of the estimation stage consisted of 10 iterations.

The  $k'$  and  $c'$  coefficients of eq. 11 were identified once for a 65Hz stimulation and they were unchanged for the other stimulation frequencies and Motor Units.

The estimation results of the muscle surface acceleration, displacement and the contraction force have been presented in fig. 2. The maximal output prediction error (E) has been below 1% for all tested signal. The estimated force signals have been also consistent with the recorded force. The contraction duration and the maximal force amplitude can be properly estimated for a single twitch (1Hz stimulation) as well as for unfused tetani contractions (40, 65Hz). For a 40Hz stimulation a low frequency disturbance is visible in the estimated force signal, however it has not been observed in the force signal recorded *in vivo*. This reveals that the estimation method is not robust for a low frequency disturbances or for accidental muscle movements.

During tetani contractions (the results have not been presented in fig. 2) the force profile and the duration of the muscle contraction have been estimated properly, although the recorded force amplitude has been bigger than this estimated from the muscle surface displacement signal. This effect has been explored previously by Orizio [5], reporting the nonlinear, exponential relationship between the muscle surface displacement and the contraction force. The more accurate results could be obtained only if the elasticity coefficient  $k'$  was a nonlinear function of the contraction force. It has been observed that during stronger contractions the low frequency disturbances of the estimated force have not been observed. This might be promising for an estimation of the voluntary contraction force.

Moreover the signals recorded during activation of the other MUs have been taken into account. The estimated force amplitudes have been different from this recorded *in vivo*, however the force profile and the contraction duration have been estimated correctly. This may be explained by the variable location of the MUs in the muscle, thus the relative positions of the microphone and the MMG sources are different, and the parameters of the eq. 11 should be identified again.



**Fig. 2.** The results of the estimation procedure performed for a 3 different stimulation frequencies of a single MU. (A-C) The recorded and the pre-processed MMG signals ( $Y$ ). (D-F) The estimated muscle surface acceleration ( $\hat{U}$ ). (G-I) The estimated muscle surface position. (J-L) The estimated ( $\hat{F}$ ) and the recorded ( $F$ ) contraction force. The estimated results are in black while the original recordings are in grey. The signals amplitudes are normalized with respect to the maximal amplitudes of the twitch contraction (1Hz stimulation).

The presented method is the first approach for the force estimation from the MMG signal recorded by the microphone. It enables to estimate correctly the contraction force amplitude and the duration. Moreover, such the method can be used on-line. For more accurate results the method should be extended to reflect a nonlinear relationship between the muscle surface displacement and the force, and to detect and filter the low frequencies disturbances.

### Acknowledgment

The work was partially supported by the Polish Ministry of Education and Science, project no. 1445/T11/2004/27.

## References

1. E. Bichler and J. Celichowski. Changes in the properties of mechanomyographic signals and in the tension during the fatigue test of rat medial gastrocnemius muscle motor units. *J Electromyogr Kinesiol*, 11(6):387–394, Dec 2001.
2. Anna Jasklska, Pascal Madeleine, Artur Jasklski, Katarzyna Kisiel-Sajewicz, and Lars Arendt-Nielsen. A comparison between mechanomyographic condenser microphone and accelerometer measurements during submaximal isometric, concentric and eccentric contractions. *J Electromyogr Kinesiol*, May 2006.
3. P. Kaczmarek, J. Celichowski, and A. Kasiski. Experimentally verified model of mechanomyograms recorded during single motor unit contractions. *J Electromyogr Kinesiol*, 15(6):617 – 630, 2005.
4. Lennart Ljung. *System Identification: Theory for the User (2nd Edition)*. Prentice Hall PTR, 1998.
5. C. Orizio, R. V. Baratta, B. H. Zhou, M. Solomonow, and A. Veicsteinas. Force and surface mechanomyogram relationship in cat gastrocnemius. *J Electromyogr Kinesiol*, 9(2):131–140, Apr 1999.
6. Claudio Orizio, Massimiliano Gobbo, Arsenio Veicsteinas, Richard V Baratta, Bing He Zhou, and Moshe Solomonow. Transients of the force and surface mechanomyogram during cat gastrocnemius tetanic stimulation. *Eur J Appl Physiol*, 88(6):601–606, Feb 2003.
7. Jean-Jacques Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall; United States Ed edition, 1990.
8. M. Watakabe, K. Mita, K. Akataki, and K. Ito. Reliability of the mechanomyogram detected with an accelerometer during voluntary contractions. *Med Biol Eng Comput*, 41(2):198–202, Mar 2003.



---

# Automatic Segmentation of EMG Signals Based on Wavelet Representation

Przemyslaw Mazurkiewicz

Institute of Control and Information Engineering  
przemyslaw.mazurkiewicz@put.poznan.pl

**Summary.** In this paper the automatic segmentation of EMG signals based on wavelet representation is presented. It is shown that wavelet representation can be useful in detecting particular spikes in EMG signals and the presented segmentation algorithm may be useful for the detection of active segments. The algorithm has been tested on the synthetic model signal and on real signals recorded with transcutaneous multi-point electrode.

## 1 Introduction

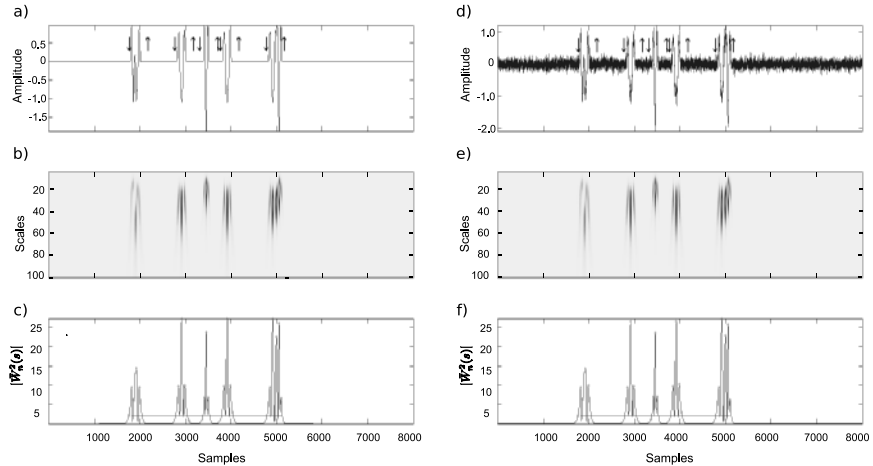
Electromyogram (EMG) plays an important role in applications related to neuromuscular systems. Based on the information carried in electrophysiological signals one may control the prosthesis of the limbs, diagnose the disorders of the neuro-muscular system and, what is the most interesting learn, how muscles fibres are controlled by motoneurons. Identification of the mechanism of muscle fibres excitations by motoneurons allows to render artificial sequence of discharges in patients with paralysis with the goal of restoring the movement. It is possible to estimate the nerve code due to the relationship between the electrical activity (EMG signal) of muscles and the discharging motoneurons. Single discharge of a motoneuron causes action potentials (AP), which propagate from neuromuscular junctions to tendons of the innervated muscles fibres. An electrode located near active motor units (MUs) records their potentials (MUAPs) summed in the time and space domain which constitute the recorded EMG signal. A sophisticated tools which decompose such a signal to components related to the particular active motor units allow to estimate the nerve code. Several decomposition systems have been proposed [1][2][3]. The first step in most decomposition systems is to cut EMG signal into segments where possible MUAPs occurred. These active segments are further used in the classification and clustering task. Segmentation methods have been implemented according to different approaches. In [3] a special 3 channel needle electrode has been used. If during contraction the amplitude of the recorded signal crossed a threshold value the acquisition with

fixed time-windows started. With this method EMG signals have been clustered online.

Another way is to cluster MUAPs offline. Example of this method is described in [1], where segmentation is based on a threshold value depending on the maximum value of a recorded signal and its mean absolute value. When the signal level crosses the threshold value a fixed length window is located with respect to the peak in a way to contain such a peak in its center. Samples in the window are considered as possible MUAP.

Basic assumption in the decomposition task is that every MU has its characteristic shape observed in the recorded EMG. Ability to distinguish between different MUAPs depends on how EMG signals are recorded. It is much easier to decompose signals recorded with the intramuscular needle electrodes because of the higher selectivity. In such a case the recorded signals include potentials from a small number of the MUs located close to the electrode. Signals recorded with the surface electrodes are much noisier and compound of spikes related to a larger number of MUs. In such signals it is very difficult to distinguish between particular MUs, especially when the force generated by the muscles fibres increases. This is the reason, why most of the developed decomposition systems work on intramuscular recordings. Moreover such systems cannot be used in FES or in rehabilitation. To overcome this restriction decomposition system should be able to decompose the surface EMG signals. The problem is that due to the mentioned properties of the surface EMG signals it is very difficult to implement such a system. Examples of such systems are described in [8],[7]. These papers present methods of the decomposition of the compound EMG signals which are not split into segmentation and classification stage. Another solution to the surface EMG decomposition may be proposed, which is similar to those, for intramuscular signals but more sophisticated methods should be applied at particular steps. These advanced methods should work well with noisy, superimposed signals, with signals where potentials shapes of particular MUs vary, its amplitude changes due to changes in the contact area between skin and electrodes. Further research should focus on the development of not only the algorithms but also on the appropriate hardware. To overcome limitations of the surface EMG segmentation, special acquisition techniques should be applied.

The author of this paper is aimed at the development of the surface EMG decomposition systems. This paper presents an algorithm of the automatic EMG surface signal segmentation which is a first stage of the decomposition system. The main part of the algorithm is an averaged in scales continue wavelet transform. In [6] it has been shown, that the averaged wavelet spectrum can be appropriate for local maxima location. With appropriate chosen mother wavelet the located maximum in wavelet representation corresponds to spikes in recorded signals. In section 2 basic assumptions and the proposed algorithm in details have been shown. In section 2.3 some examples of a synthetic signal and real signal segmentation have been shown.



**Fig. 1.** Synthetic EMG test signal segmentation. a), d) generated signal with marked begin and end of active segments noiseless and noisy signal with variance  $\sigma^2 = 0.05$  respectively (downarrow and uparrow respectively), b), e) corresponding signals wavelet spectrums, c), f) corresponding averaged wavelet spectrums with marked maxima (dots).

## 2 Automatic Signal Segmentation - Algorithm

### 2.1 Basic Assumptions

#### Signals

Let  $x(k) = [x_1, x_2, \dots, x_n]$  describes a vector of discrete EMG signal samples being processed. To demonstrate the important feature of the proposed segmentation algorithm a synthetic test signal has been designed. It has been assumed that signal corresponds to the sum of the potentials of 2 different MUs. Additionally, potentials of the same MU have similar shape however different to other MU potentials. For further tests a real EMG signals have been used. The signals have been recorded with surface mounted electrodes located over *tibialis anterior*, bandpass filtered with a 20-1000 Hz filter and sampled at 2500 kHz. The shape of MUs potentials vary from evident spike potentials to a wave corresponding to multiple phase signals (3-4 phases).

#### Signal Processing Tools

As a basic processing instrument supporting the segmentation task of the EMG signal a wavelet representation of signals have been proposed. The main virtue of that representation is that it easily, regardless of the noise level determines potentials in a compound signal. Wavelet representation is defined in eq. (1)

$$W_n(s) = \sum_{k=0}^{N-1} x(k)\Psi\left(\frac{k-n}{s}\right) \tag{1}$$

where  $\Psi\left(\frac{k-n}{s}\right)$  is a shifted and scaled version of a mother wavelet. Mexican hat wavelet have been chosen. It has been shown in previous paper that such a wavelet is suitable to identify peaks in potentials.

During segmentation task the information about location of the peaks in time is searched for. Due to the 2D parameter space of the wavelet representation averaging with respect to the scales of the wavelet spectrum have been proposed according to eq. (2) [5].

$$\bar{W}_n^2(s) = \sum_{j=1}^J |W_n(s_j)|^2 \tag{2}$$

where  $j$  stands for the scale index. such an averaged wavelet spectrum reflects signal energy fluctuations in time.

### 2.2 The Algorithm

The proposed algorithm consists of the following steps:

1. Evaluation of  $W_n(s)$  scale-time representation of the signal  $x(n)$  according to eq. (1)
2. Averaging of the wavelet spectrum with respect to the (2)
3. Finding local maxima of (2) passing over a threshold  $\beta$
4. Putting  $L$ -sample-length window at the  $m$ -th local maximum, where  $m = [1, 2, \dots, M]$  and  $M$  is the number of detected local maxima.
5. If the  $m$ -th maximum is located within a range of  $(\gamma - \delta, \gamma + \delta)$  then samples of  $x(n)$  would be attributed to an active segment  $\omega_i(k)$  as follows:

$$\omega_i(k) = x([m - 0.5 * L : m + 0.5 * L]) \tag{3}$$

where  $\gamma$  is a mid point of a window, and  $\delta$  - a range of samples round the mid point.

The window length is expected to be length enough to captures all major potentials from the EMG signals. From the physiology it is known that potentials duration is up to 20 ms in healthy patients.

Additionally, the threshold  $\beta$  have been used to eliminate small energy components which are regarded as a noise.

### 2.3 Results

#### Synthetic Signal Segmentation

The generated test signal is shown Fig.1a,d. One can see 5 action potentials, where 2 of them clearly belong to the first MU (round 3000th and 4000th sample)

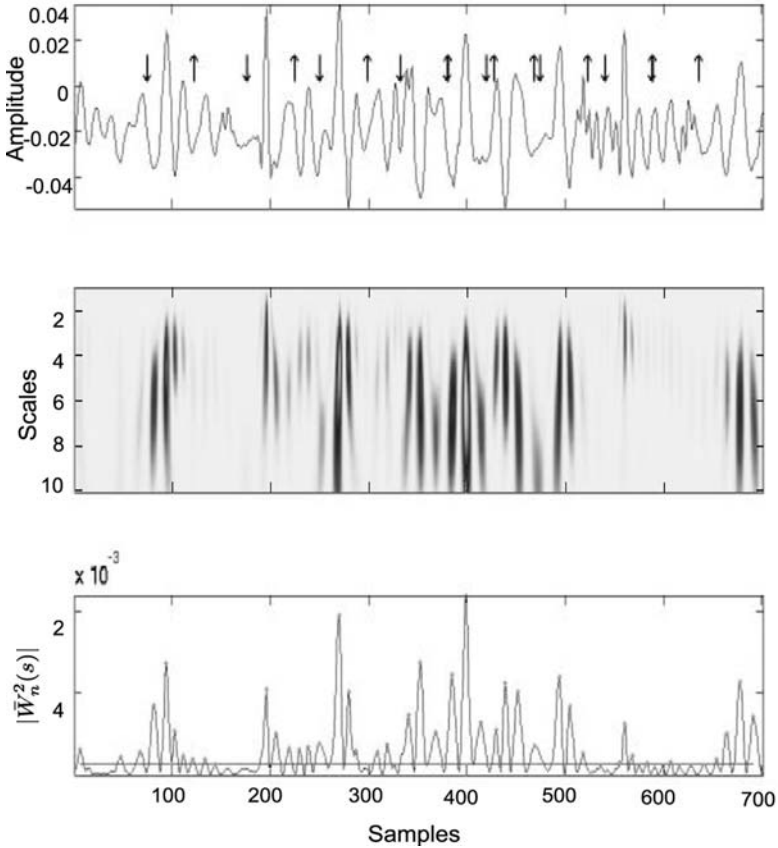
and the action potential in the round of the 3500th sample aligned at the second MU. Two apparent potentials (around 2000th and 5000th sample) are summed potentials of both active MUs. In Fig.1b,e a wavelet spectrum of the analysed test signal is given, and in Fig.1c,f an wavelet spectrum averaged in scale is presented. The important remark is that energy fluctuations are much higher in the round of apparent action potentials. The highest peak corresponds to the best correlation between potential and a wavelet and can be regarded as spike location indicator. This kind of indicator can be especially helpful while detecting spikes in noisy signals, what can be seen by comparing Fig.1a,c and Fig.1d,f. Results of automatic segmentation can be seen in Fig.1. The analysed signal has been cut automatically into 5 segments. Segments #2,#4 belong to the first MU, segment #3 to the second MU and segments #1 and #5 are due to the sum of action potentials of active MUs. In Tab.1 the position of segments position in time have been evaluated as a function of a additive noise variance. According to this table segmented potentials minimally differ from the reference noiseless segments.

**Table 1.** Ranges of active segments of the synthetic signal for three different noise variances:  $\sigma_1^2 = 0$ ,  $\sigma_2^2 = 0.01$ ,  $\sigma_3^2 = 0.1$

#seg	$\sigma_1^2$	$\sigma_2^2$	$\sigma_3^2$
1	1721:2121	1722:2122	1720:2120
2	2717:3117	2717:3117	2717:3117
3	3257:3657	3257:3657	3256:3656
4	3717:4117	3717:4117	3717:4117
5	4717:5117	4717:5117	4717:5117

### The Segmentation Results of the Real EMG Signal

In this example a real EMG signal have been segmented with the proposed method. The recorded real signal is presented in Fig.2a. The algorithm subdivided the signal into 9 active segments. In this case 19.6 ms window length have been chosen to segments compound signal. A threshold have been set to eliminate small energy components. looking at Fig.2a one may notice a very similar pairs of segments (#2 with #8 and #4 and #6). The segment #4 includes small artefact. There are visible some other superimposed potentials like #7 and #9. All these segments have been detected based on the information contained in the averaged wavelet spectrum (Fig.2c) by the proposed segmentation algorithm.



**Fig. 2.** Real EMG signal segmentation. a) recorded signal with marked begin and end of active segments (downarrow and uparrow respectively), b) analysed signal wavelet spectrum, c) averaged wavelet spectrum in scale with marked located maxima (dot).

### 3 Conclusions

From the above presented results it is clear that the described segmentation method is a promising tool. It is worth to noticed that used wavelet spectrum averaged in scale can be suitable for the spike detection especially in noisy signal.

With the wavelets analysis there is always an open question how to choose a mother wavelet. In a case of EMG signals it is rather difficult to find a wavelet with the shape similar to MUAPs because of the large variability of their shapes. It is rather better to find a wavelet which allows to locate local extrema of a particular MUAPs. Such extrema may be used as a reference points of MUAPs as it have been shown in this paper. Additionally this reference points may be used for further steps as characteristic points indexing the segments. In real applications a problem may occur due to of some energy artifacts such as movement

generated artifacts or some other. However these artefacts can be well localized by the wavelet function and can be extracted by the algorithm as an active spurious segment. This is a potential source of errors in the further signal classification steps. These steps will have to eliminate this artefact but keeping a real MUAP. At the moment this kind of interference has to be avoided by careful acquisition of signals to be analysed.

It has been shown that the presented method is useful in the surface EMG signals segmentation. Further tests on larger groups of the test signals are required.

## Acknowledgment

The work was partially supported by the Polish Ministry of Education and Science, project no. 1445/T11/2004/27.

## References

1. C. I. Christodoulou, C. S. Pattchis (1999) Unsupervised Pattern Recognition for the Classification of EMG Signals, *IEEE Transactions on Biomedical Engineering* 46:169 – 178
2. M. H. Hassoun and Ch. Wang and R. Spitzer (1994) NNERVE: Neural Network Extraction of Repetitive Vectors for Electromyography - Part I: Algorithm, *IEEE Transactions on Biomedical Engineering* 41: 1039–1052
3. R.S LeFever, C. J. DeLuca (1982) A procedure for decomposing the myoelectric signal into its constituent action potentials: part I, execution and test for accuracy. Technique, theory and implementation, *IEEE Transactions on Biomedical Engineering*, 29:149–157
4. D. W. Stashuk (1998) Decomposition and quantitative analysis of clinical electromyographic signals, *Medical Engineering & Physics* 21:389 – 404
5. C. Torrence and G. P. Compo (1998) A Practical Guide to Wavelet Analysis, *Bulletin of the American Meteorological Society* 79:61–78
6. D. Nishikawa and W. Yu and H. Yokoi and Y. Kakazu (1999) Analyzing and discriminating emg signals using wavelet transform and real-time learning method, *Intelligent Engineering Systems Through Artificial Neural Networks* 9:281–286
7. Yamada, R. and Ushiba, J. and Tomita, Y. and Masakado, Y. (2003) Decomposition of electromyographic signal by principal component analysis of wavelet coefficients, in *IEEE EMBS Asian-Pacific Conference on Biomedical Engineering* 118–119
8. H. Nakamura and M. Yoshida and M. Kotani and K. Akazawa and T. Moritani (2004) The application of independent component analysis to the multi-channel surface electromyographic signals for separation of motor unit action potential trains: part I-measuring techniques, *Journal of Electromyography and Kinesiology* 14:423–432

---

# Elimination of Linear Structures as an Attempt to Improve the Specificity of Cancerous Mass Detection in Mammograms

Marcin Bator and Leszek J. Chmielewski

Institute of Fundamental Technological Research, Polish Academy of Sciences,  
Świętokrzyska 21, PL 00-049 Warsaw  
{mbator, lchmiel}@ippt.gov.pl

**Summary.** In the screening mammographic examination of large parts of populations thousands of mammograms are analysed. The Computer Aided Diagnosis methods available still tend to produce too many false positive (FP) detections with respect to the number of true positive (TP) detections, which makes it impractical to use such methods to support the human observer in the analysis of mammograms. In this paper an attempt has been made to decrease the number of FP errors in the hierarchical correlation-based cancerous mass detection method by eliminating the images of linear structures (LSs) from the mammograms. The LSs were detected with an accumulation-based line detector and the image intensity function in the regions of the LSs was interpolated with an anisotropic membrane. Examples of images representing typical detection problems caused by the LSs selected from the MIAS database suggest the feasibility of the proposed approach.

## 1 Introduction

Breast cancer is an important social problem. It can be estimated that in the US in one out of eight women the breast cancer will develop at some point during her life [11]; the corresponding number for Poland is one in 16 women [6]. Early detection of breast cancer makes it possible to apply a sparing treatment and elongates the survival. Therefore, screening the groups of elevated risk (women above the age of 50) is justified, which involves the analysis of thousands or millions of mammograms. The development of Computer Aided Diagnosis (CAD) methods might help the radiologists in carrying out this task. In less than 10% of women the result of the screening mammography is positive, that is, it suggests further investigation. If a CAD system could classify at least a part of 90% of mammograms as negative prior to the investigation by a human, it would greatly reduce the workload of radiologists and make it possible for them to pay more attention to the remaining mammograms, potentially carrying the features of an abnormality. A CAD system, to be useful in improving the performance of a radiologist, should have a sufficiently large specificity at a given sensitivity



level, comparable to that of a human observer. In [17] a discussion has been presented which leads to the conclusion that there should be not more than 1.5 false positive detections (FP) for each true positive detection (TP). In the present paper we report our attempt to increase the TP/FP ratio in a cancerous mass detection algorithm described in [1] by eliminating the linear structures from the mammograms.

Comprehensive studies of all aspects of breast cancer imaging can be found in [8]. Reviews of cancerous masses detection and classification can be found in [11] and to some extent also in [15]. Linear structures (LS) [7, 16], also called curvi-linear structures [10, 12], are the blood vessels, milk ducts and connective tissue fibers (Cooper ligaments) in normal mammograms, and the spicules in pathologic mammograms. Some authors report that their classification should improve the subsequent detection and classification of the features of cancer, like in [16]. Others, like in [12], state that the attenuation of the LSs improves the results of analysis. The LSs related to cancerous masses are spicules; other LSs interfere in the detection of masses. In this study a mass detection algorithm using no information on the spicules is used, so we go along the line of elimination of all the LSs without their classification.

## 2 Detection of Cancerous Masses

The cancerous masses were detected with a multiscale template matching algorithm reported previously in [1]. A template is an auxiliary image representing the model brightness distribution in a mass. The template is shifted across the image and the similarity measure between the template and the corresponding window of the image is calculated. As the similarity measure the correlation coefficient was used, as expressed by the equation

$$c(T, I) = \frac{\sum_{j=1}^N (t_j - \bar{t})(i_j - \bar{i})}{\sqrt{\sum_{j=1}^N (t_j - \bar{t})^2} \sqrt{\sum_{j=1}^N (i_j - \bar{i})^2}}, \quad (1)$$

where  $T$  is a template and  $I$  is a window in the image, both containing  $N$  pixels numbered consecutively,  $t_j$  is a grey level in a template pixel,  $i_j$  is a grey level in a window pixel, and  $\bar{t}$ ,  $\bar{i}$  are the mean grey levels of the template and the window, respectively. The correlation coefficient belongs to an interval  $(-1, 1)$  and depends only on the relation of shapes of the template and the window, not on their grey scale. A template with a hemispherical brightness was used:

$$t_R(x, y) = \sqrt{R^2 - x^2 - y^2} \text{ for } x^2 + y^2 \leq R^2, \quad (2)$$

where  $x, y$  are pixel coordinates in a local coordinate system of the template having origin at the template centre. As in [1], the the Fast Fourier Transform was used and the correlation was calculated in the Fourier domain, while unlike in [1], the template was circular.

As advised in [14] and according to the previous experience [1], the template matching combined with a multiscale approach was used. The templates with 8 radii from 36 to 108 pixels, differing by a factor of  $\sqrt{2}$ , were used. To reduce the amount of calculations, images in four scales were analysed: 1:1, 1:2, 1:4 and 1:8, with two radii in each: 36 and 51 pixels. Templates with such radii detect objects having radii of 30-43 pixels and 43-60 pixels, respectively. The multiscale approach was designed according to the observation made in [9] that instead of using a template with two times larger radius one could use an image with two times smaller linear dimensions, obtaining approximately the same results. Combining such a resolution pyramid with FFT calculations mentioned above reduced the calculation time for a single mammogram to less than 10 min on a 2 GHz computer. At the image resolution of  $50 \mu\text{m}/\text{pix}$ , the masses of diameters from 3 to 48 mm could be detected.

The final detection result was calculated by thresholding the resulting correlation maps at the subsequent pyramid levels with a single threshold and aggregating the partial results received onto the full scale level 1:1.

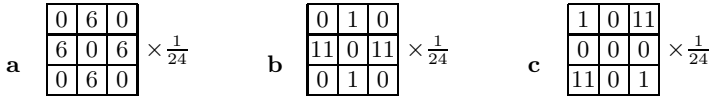
### 3 Linear Structures

#### 3.1 Detection

The method used for detection of the linear structures was the evidence accumulation-based line detection algorithm already described in sufficient detail in [4, 5], and previously in [3]. In the present application, linear structures of width between 2 and 16 pixels were sought. Structures having the line intensity maxima smaller than 0.1 of the maximum intensity of the strongest LS in the given image were neglected and the ridges of the remaining, stronger structures were followed (some less important details were omitted for the sake of compactness; for details see [4], p. 360 or [5], Chapt. 6.3.6 and 6.8).

The line detector used finds the line width  $w_0$  as the distance between the loci of maximum image intensity gradients, as is typically done by detectors. However, the actual width  $w$  of the LS in the image is larger. Therefore, the width found was multiplied by a factor slightly larger than one:  $w = 1.1 w_0$ .

The primary results of the detection process for each mammogram were the binary masks of the LSs. In the LS elimination process described in the following Section, also the the map of the directions of the LSs and the relative distance transform, defined further, were used. These were easy to generate as the byproduct of the detection process, due to that each detected pixel is related to its respective pixel in the skeleton of the line, in which the local line width and direction has been stored by the line detection algorithm. The direction is simply copied from that stored in the relative skeleton pixel. The relative distance transform in a given pixel of the LS is the ratio of its distance from the line edge to the distance of its relative skeleton pixel from the edge, equal to a half of the width.



**Fig. 1.** Examples of masks used for interpolation in the membrane spanning algorithm. The sum of elements is divisible by four to make it easier to construct the isotropic mask. (a) Isotropic membrane; (b) membrane stronger in the W-E direction; (c) membrane stronger in the SW-NE direction.

### 3.2 Elimination

A linear structure in the breast projects itself on the mammogram together with other tissues and manifests itself as a bright object in the image. The image intensity function of a mammogram can be split as follows

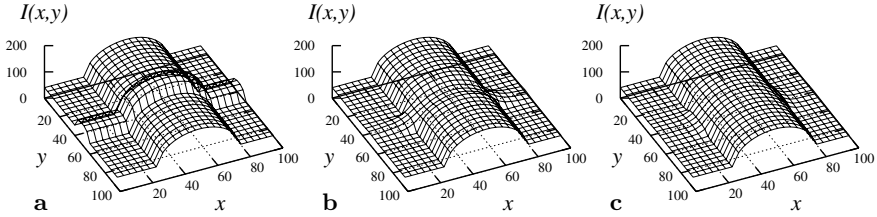
$$I(x, y) = I_0(x, y) + I_{LS}(x, y), \tag{3}$$

where  $I$  is the image intensity of the mammogram,  $x, y$  are coordinates,  $I_0$  is the intensity without the LS, and  $I_{LS}$  is the intensity of the LS alone.

The possible approaches to the elimination of the LSs are: 1° subtraction of the  $I_{LS}$  component of the intensity from the image, 2° replacement of the image intensity in the regions occupied by the LSs with the intensity  $I_0$ , and 3° suppression of the component  $I_{LS}$  so that the resulting intensity is closer to  $I_0$ . The first approach necessitates for the model of the LS. The images of the LSs are very different so this approach seems unrealistic. The second approach makes it necessary to know  $I_0$  in the regions of the LSs. In the third approach it is assumed that the method of suppressing the LSs is known. For example, in [12] the LSs were attenuated by averaging the image intensity in the regions occupied by the LSs in a  $11 \times 11$  pixels window.

In this paper the second approach was used. The image intensity  $I_0$  without the LSs in the regions of the LSs was estimated as the interpolation of the image intensity from outside the regions occupied by the LSs. This was done by spanning a *membrane* across the regions of the LSs. An iterative algorithm was used, as described in [2], Chapter 6.2.1. In the present application that algorithm becomes radically simplified to finding the averages of the neighbouring pixels, with a 4-neighbourhood. This corresponds to the iterative convolution of the image with the mask shown in Fig. 1a, until the result stabilizes. The intermediate calculations were performed in the floating-point arithmetic. The method was similar to that used in [12] in that the convolution equivalent to averaging was used, while the main difference was that the convolution was repeated until the surface received stabilized.

Spanning an isotropic membrane across the regions occupied by the LSs yields undesirable results where the linear structure crosses the regions having different signs of the curvature, that is, the “hills” and “valleys”, as shown in Fig. 2a and b. To reduce the saddles near the ridges and bottoms an anisotropic membrane was used. The direction of larger stiffness was across the LS, as found by the line



**Fig. 2.** Using the anisotropic membrane to interpolate an LS crossing a “hill” in the image intensity function. (a) Original image intensity: detected LS along  $Ox$ , ‘hill’ along  $Oy$ ; (b) LS eliminated with the isotropic membrane: saddles present in the ridges and bottoms of the intensity graph; (c) LS eliminated with the anisotropic membrane, stronger in the direction across the LS: saddles greatly reduced.

detector. These directions were rounded to four characteristic directions differing by  $45^\circ$ , and for each pixel in the LS the appropriate mask was used. The examples of the masks are shown in Fig. 1b and c and the result of using this approach can be seen in Fig. 2c. The ratio of the stiffnesses in the perpendicular directions was close to 10, which was enough to receive a clear improvement in relation to the described effect.

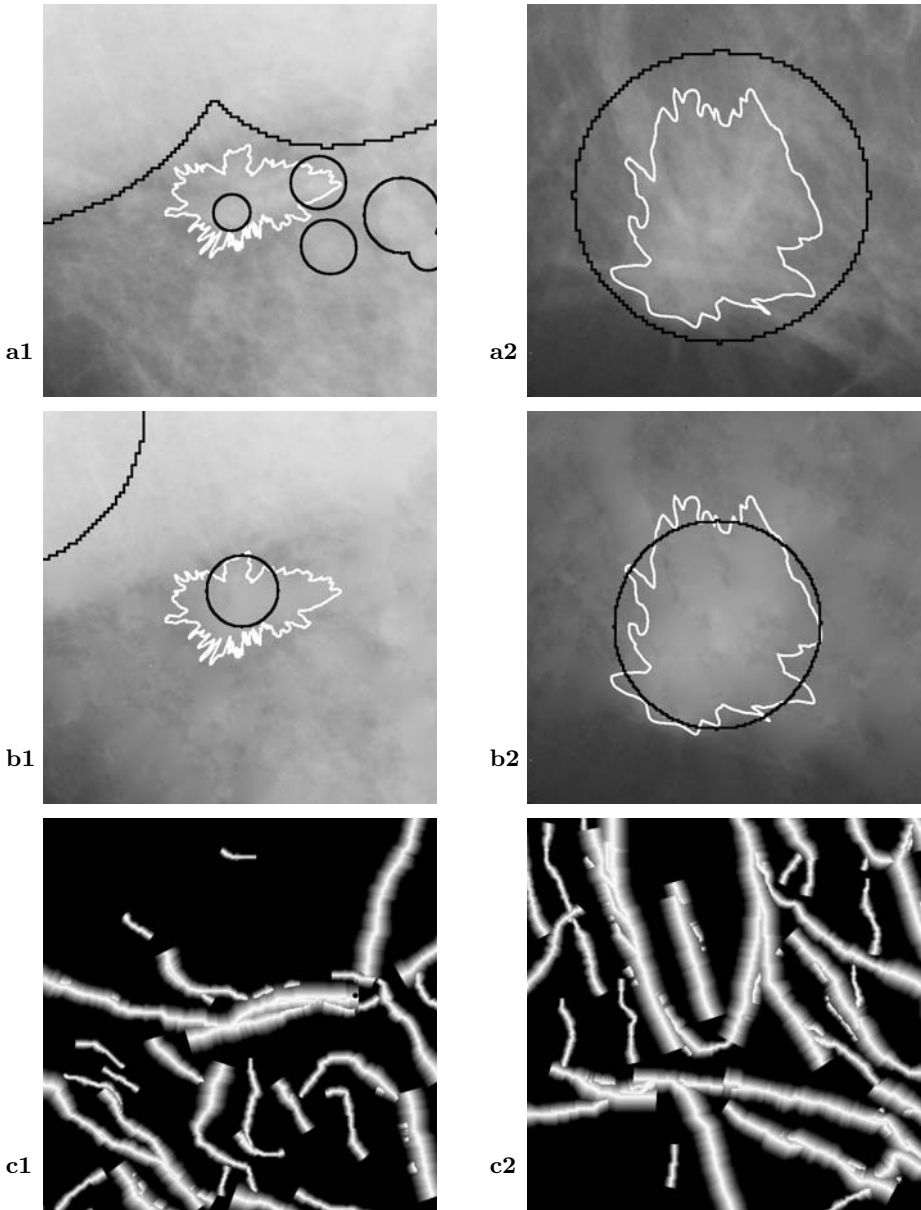
In the final algorithm each interpolated surface was inserted into the image intensity function with a narrow transient stripe in which the two functions were gradually interchanged by linear weighting. The weighting functions used could be easily calculated from the relative distance transform generated by the line detection algorithm. The stripe width corresponded to the difference between  $w$  and  $w_0$ , mentioned above.

According to the multiscale organisation of the correlation-based mass detector, the line elimination process was also performed in four scales. However, the line detection was done only in the scale 1:4 and its results were appropriately recalculated to the remaining three scales.

## 4 Examples of Results

The feasibility of the method has been tested on the images from the MIAS database [13]. The results for two images have been selected for presentation due to that the correlation method yields unsatisfactory results in these images because of linear structures passing over cancerous masses. Detection results before and after the elimination have been shown in Fig. 3 and Table 1. The detected masses have been marked with circles according to the shape of the templates used. The radii of the circles correspond to the radii of the masks, recalculated to the full scale 1:1, for which the maximum correlation reached the threshold. Thresholds used were equal to correlations shown in Table 1.

It can be observed that in the examples shown: 1<sup>o</sup> the correlation after the elimination of the LSs grew, 2<sup>o</sup> the number of false positive detections decreased (effect visible in the window shown only for `mdb124rm`), and 3<sup>o</sup> the location



**Fig. 3.** Examples of results – windows  $800 \times 800$  of mammograms from the MIAS database: (1) `mdb124rm`, upper left corner at (1340, 24); (2) `mdb264rm`, upper left corner at (2156, 836). (a) Original images; (b) images with the LS eliminated; (c) relative distance transforms of the LSs. The *black circles* indicate the detected cancerous masses, in the original and processed images, respectively, and the *white contours* indicate the shape of the masses marked by a human observer.

**Table 1.** Change of parameters of the masses in the mammograms shown in Fig. 3

mammogram	before elimination		after elimination	
	correlation	radius [pix]	correlation	radius [pix]
mdb124rm	0.63	36	0.76	72
mdb264rm	0.76	288	0.80	204

of the detected mass became closer to that determined by a human observer. A cautious conclusion can be drawn from these observations that it could be possible to increase the threshold used in the detection of cancerous masses, which in effect should reduce the number of false positive detections at a given level of true positive detections, thus improving the relation of specificity to sensitivity of the method.

## 5 Conclusion and Further Work

To reduce the number of false positive detections while maintaining the number of true positive detections in the cancerous mass detection method applied to mammograms, the elimination of linear structures was applied. The mass detection method used was the hierarchical correlation-based detection described previously in [1]. The linear structures were detected with the accumulation-based line detector [3, 4, 5] and the elimination was performed by interpolation of the image intensity in the regions of the LSs with an anisotropic membrane being a simple derivation of the model described in [2].

The results for images representing typical detection problems caused by the LSs passing over cancerous masses, selected from the MIAS mammographic database [13], suggest the feasibility of the proposed approach. The trials with a large set of benchmark mammographic images are necessary to validate the proposed method. Preliminary results of such trials being carried out at present with the entire MIAS database are promising.

**Acknowledgement.** The research was financed by the Ministry of Education and Science as the research project No 3 T11C 050 29 in 2005-2008.

We thank dr Ewa Wesołowska from the Center of Oncology, Warsaw, for providing the contours indicating the shapes of cancerous masses shown in Fig. 3.

## References

1. Bator M, Nieniewski M (2006) The usage of template matching and multiresolution for detecting cancerous masses in mammograms. In E. Piętka et al (eds), *Proc. 11th Int. Conf. Medical Informatics & Technology*, Wisła-Malinka, Poland, 25-27 Sept 2006, pp 324–329
2. Blake A, Zisserman A (1987) *Visual Reconstruction*. MIT Press, Cambridge, MA, London

3. Chmielewski L (2006) Detection of non-parametric lines by evidence accumulation: Finding blood vessels in mammograms. In K. Wojciechowski et al (eds) *Computer Vision and Graphics: Proc. Int. Conf. Computer Vision and Graphics ICCVG 2004*, volume 32 of *Computational Imaging and Vision*, Warsaw, Poland, 22-24 Sept, 2004. Springer, Berlin Heidelberg New York, pp 373–380
4. Chmielewski L (2005) Specification of the evidence accumulation-based line detection algorithm. In Kurzyński M., Woźniak M, Puchała E et al. (eds). *Proceedings of international conference on computer recognition systems CORES 2005*, Rydzyna, Poland, 22-25 May 2005. Volume of Advances in Soft Computing. Springer, Berlin Heidelberg New York, pp 355-362
5. Chmielewski LJ (2006) *Metody akumulacji danych w analizie obrazów cyfrowych*. Akademicka Oficyna Wydawnicza EXIT, Warszawa
6. Dziukowa E, Wesolowska E (eds) (2006) *Mammografia w diagnostyce raka sutka*. Medipage, Warszawa, second edition
7. Hong B-W, Brady M (2003) Segmentation of mammograms in topographic approach. In *Proc. IEE Int. Conf. on Visual Information Engineering VIE'03*, Guildford, U.K., July 2003
8. Kopans DB (1998) *Breast Imaging*. Lippincott-Raven, Philadelphia
9. Liu S, Babbs CF Delp EJ (2001) Multiresolution detection of spiculated lesions in digital mammograms. *IEEE Trans Image Proc* 10(6):874–884
10. Rangayyan RM, Ayres FJ (2006) Gabor filters and phase portraits for the detection of architectural distortion in mammograms. *Med Bio Eng Comput* 44(10):883–894
11. Sampat MP, Markey MK, Bovik AC (2005) Computer-aided detection and diagnosis in mammography. In Bovik AC (ed) *Handbook of Image and Video Processing*. Academic Press, pp 1195-1217
12. Sheshadri HS, Kandaswamy A (2005) Detection of breast cancer tumor based on morphological watershed algorithm. *ICGST Int J Graph Vision and Image Proc* 05(V5):17–21
13. Suckling J, Parker J et al (1994) The Mammographic Images Analysis Society digital mammogram database. In: Gale AG, Astley SM et al (eds) *Digital Mammography. Excerpta medica international congress series*, vol 1069, pp 375-378
14. te Brake GM, Karssemeijer N (1999) Single and multiscale detection of masses in digital mammograms. *IEEE Trans Med Imaging* 18(7):628–639
15. Thangavel K, Karnan M et al (2005) Automatic detection of microcalcifications in mammograms – A review. *ICGST Int J Graph, Vision and Image Proc* 05(V5):31–61
16. Zwiggelaar R, Astley SM et al (2004) Linear structures in mammographic images: detection and classification. *IEEE Trans Med Imaging* 23(9):1077–1086
17. Zwiggelaar R, Parr TC et al (1999) Model-based detection of spiculated lesions in mammograms. *Med Image Anal* 3(1):39–62

---

# Automatic Counting of Neural Stem Cells Growing in Cultures

Anna Korzynska

Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences, ul. Ksiecia Trojdena 4, 02-109 Warszawa, Poland  
Anna.Korzynska@ibib.waw.pl

**Summary.** Stem cells are a potential source of cells for use in the regenerative medicine. Automation of monitoring and analysis is crucial for reliable and fast optimization of culturing methods. The goal of the first step in this investigation is to find a method of automatic cell counting on microscopic static images. In our method, an image is divided into two types of regions: the regions covered by cells and the background regions. Next the cell regions are classified into three categories: converged cells, the flatten cells and the transitional cells regions. For each type of region, the adjusted procedure estimates a quantity of cells. The quantity of cells in image has been obtained for randomly chosen images from certain sequences captured from cells which growth has been monitored using laser scanner confocal microscopy. The results of the automatic cell counting are compared with results obtained by an operator and the difference has been admissible. When a lot of frames and cells are counted, the accuracy of the proposed method has been similar to the accuracy of an expert.

## 1 Introduction

Stem cells are a potential source of cells for use in the regenerative medicine. Automation of monitoring and analysis is a crucial step for solving problem of optimization of culturing methods. Continuous monitoring of living cell cultures seems to be faster in culture condition validation than using traditional laboratory methods, such as immunohistochemistry and cytometry investigation or visual evaluation by an operator.

The goal of this step of investigation is focused on the automatic method of counting cells based on the single image. Information about changes of the cell quantity in time will be used in the next step of investigation to verify the prepared model of cell culture development.

The model includes the rates of cell proliferation and cell death. The cell proliferation noticeably increases the cell quantity, till they fulfill the plate up, while apoptosis and necrosis (rarely observed) lead to decrease of the cell quantity. Cell death or proliferation kinetics can be measured by counting a number of cells in first frame and by tracking the cells in consecutive frames. So the cell quantity in each frame is that what this study concentrates on.



Going through papers describing the quantification of the dynamics of cell colonies there are two main approaches, the direct and the indirect ones.

The direct approach uses the cell segmentation and detection according to the cell features. Mainly the following features are used:

- shape [1, 2, 3, 4]; enumeration and detection of proliferating bone marrow progenitors, cancer cells, neutrophils or artefact in smears;
- texture [5, 6]; identify nuclei by characterizing the chromatin structure;
- color [7, 8]; nuclei in fluorescent-stained samples.

The direct approach to estimation of the cell quantity requires the image segmentation, which is generally a difficult problem of the image processing and analysis. The main difficulty for the single cell segmentation is caused by the cell aggregation, what hinders the cell contours, and by the fact, that some cells are very flatten and transparent.

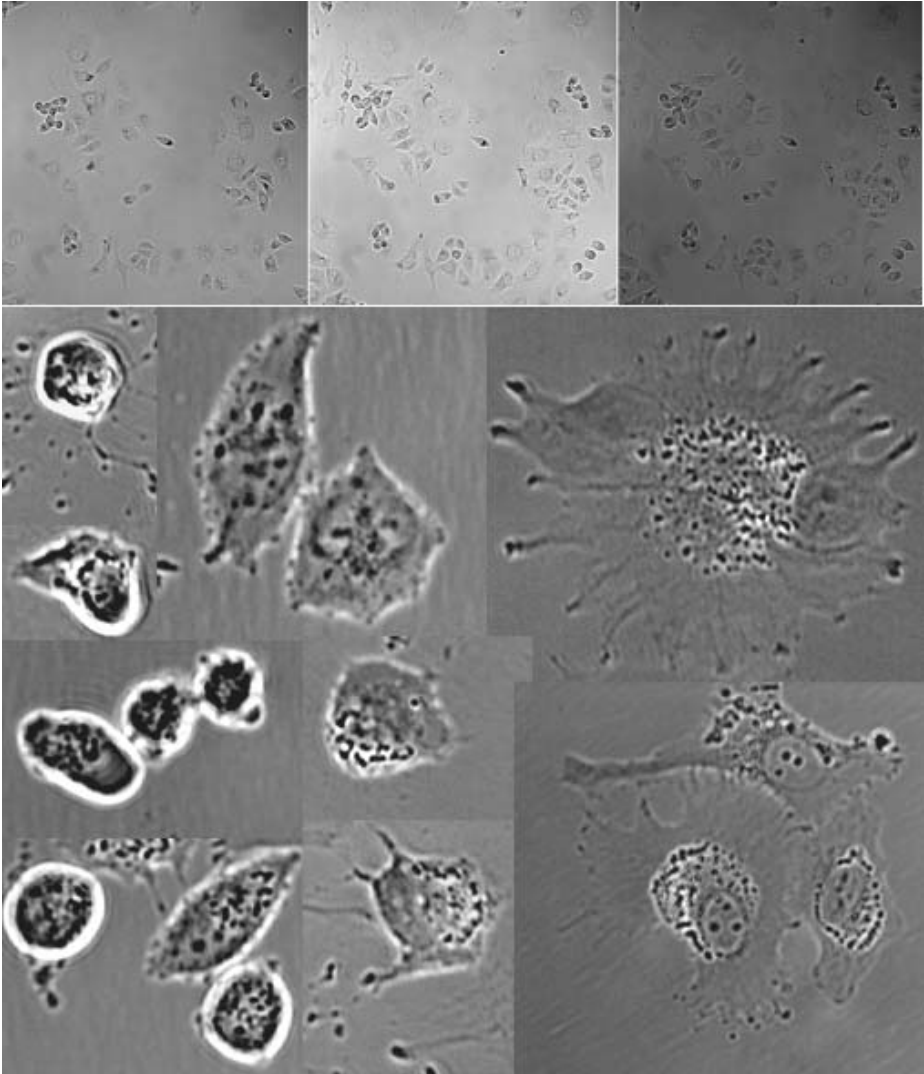
For a long time, the problem of microscopic images of living cells segmentations has been solved by detection of edges or regions [9]. Nowadays there are two directions of developing progress in this field: (1) using a priori knowledge and models [10], (2) employing the cooperation between methods, and their adaptation to the local situations in the image [11]. Within the second direction there are two methods going back to the edge and region approach: - combined texture and edge based method, proposed by the author and her collaborators [12] and - cooperative system [13].

The indirect approach to the problem of counting cells uses globally estimated features as intensity of density or density of color to approximate the cell quantity. In [14], a system to measure the relative cell quantity in culture plates makes use of total fluorescence after background fluorescence reduction as a measure of a number of cells per plate. In [7], cell quantity is estimated by dividing a cell cluster area by means of a cell area.

This paper presents an automatic method developed for cell counting employing a texture and dividing an image in homogenous regions containing similar cells. Then each cell cluster is allotted to the type of region and analyzed using the appropriate method adjusted to the type of cell in it. The result of this method and the manual counting are compared with.

### 1.1 Characteristics of Image Sequence and Cell in Images

The problem of image quality is important but very rarely addressed in publications. The quality depends on a type of microscope, illumination at the moment, resolution of acquired image [15, 16]. Laser scanning microscopy, very sensible for external illumination and with internal inhomogeneous light distribution, was used in the study [17]. A certain focus instability in a long term observation is an inherent feature of the microscopy. To observe all of these features see Fig. 1, where samples of 3 images (1st, 23rd and 46th) acquired in a single sequence with 15 minute time increment are shown in upper row. There are neural stem cells in these images and some examples of enlarged cells under samples images.



**Fig. 1.** Three images from the sequence that monitored the stem cell culture growth and the examples of cell morphology

The segmentation of cell images seems to be difficult [18] due to: the contrast quality, the deformable structure of a cell, the cell aggregation and a temporary loss of the focus. The contrast between the cells and the background is low and the histogram of such an image is unimodal. Presented farther on Fig. 2h shows histograms of different image fragments. The intensity of the background is not uniform across the image, due to the external and source light variation, Fig. 1. Significant local intensity variation across the image plane is observed

both within the cell and the background. Detection of the cell nucleus is difficult (flattened cell) or even impossible (small converged cell). Deformable and non-rigid cell structure causes that there is no existing model of the cell movement.

Appearance of the cells observed in our images varied: the cells differed in size, shape, optical and texture properties. It was impossible to find uniform criteria of the various morphology cell detection. The next step towards determination of a number of cells needs introducing the cell distribution. Cells were divided into three types: **(1)** the converged cells (bottom path on Fig. 1), **(2)** the flattened cells (upper path on Fig. 1) and **(3)** the cells starting to develop pseudopodia and elongate, but still being far from the flattened cells below called transitional cells (middle path on Fig. 1). It should be stressed that there is no strict definition of the cell types. The cell morphology evolves in time and the cell shape may simultaneously have properties of the two different types.

The proposed cell distribution has nothing to do with the cell classification in biology, however, there is some correlation between cell's physiological behavior and its appearance. The cell just before division becomes small, rounded and converged, while the cell starting to develop pseudopodia becomes rather elongated but still is far from being flattened. The flattened cell looks as very large, with/or without pseudopodia, with respectively well detectable texture around its nucleus.

Among living cells there are cell fragments within the field of vision; they remain and become visible e.g. after cell apoptosis (rectangles in Fig. 2), their area is not significantly smaller than of a rounded cell. However, their irregular shape distinguishes them from the small rounded cells. The cell fragments can be attached to the living cells or float freely in the medium.

The cells in the image are composed of the not always well distinguishable zones (Fig. 1):

- **Nuclei zone** - well visible in the flattened cell and sometimes in the transitional one. Nuclei are clean rounded structures with one or several dots inside;
- **Endoplasmatic reticulum zone** - well visible in the flattened cell and sometimes in the transitional cell. Hardly textured and hard to delineate region partly or fully around the nuclei;
- **Cytoplasm zone** - variously visible in various types of the cells. In the flattened cell it is almost clear, slightly darker than its background, with some irregular and randomly dispersed dots (cells organelles). Cytoplasm of the converged cell is remarkably darker than its background, with irregularly but closely and densely dispersed dots. While the transitional cell's cytoplasm is contained somewhere between these two stages;
- **Pseudopods zone** - well visible in the flattened cell and sometimes in the transitional one. Pseudopods are cytoplasmic extensions of the cell, which are poorly contrasted with the background, but sometimes well visible because of halo around them;
- **Background zone** - visible among the cells and cell colonies. Its intensity is nonuniform across the image but locally it can be treated as uniform;

- **White halos zone** - around the cell and cell clusters - well visible in the regions where the cells are high, what in converged and transitional cells is usually observed.

The goal of the study was to examine the cell quantity changes in time in the culture based on sequence of images. To do this the cells of each types were counted separately.

## 2 Methods and Materials

In the proposed method the cell counting is performed in two steps. In the first step, called the segmentation step, an image is partitioned into several regions: covered by clean background fragments or covered by one cell or cell cluster. In each region covered by the cells the texture is examined to divide it into the homogenous regions, if it is necessary. The second step is performed in each homogenous region separately by the appropriate counting procedure. Then the total number of cells in the image is calculated.

### 2.1 Segmentation and Cells Counting

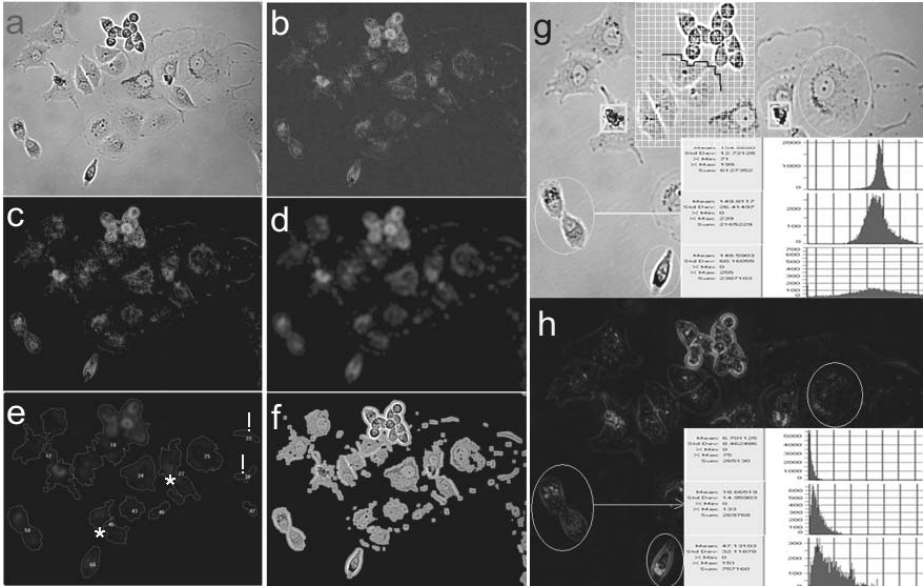
The proposed segmentation method consists of combination of some well known methods [19], which cooperate on various levels to adjust the action to the local situations in the image.

*Background suppression.* Because the mean gray level of background and the contrast of the cell against the background vary according to the camera settings, the image illumination and the region, and the background Fig. 2a, the suppression method is used. The homogeneous black background is achieved using the subtraction of the images constructed as open and close morphological operations of the gray scale images, Fig. 2b. The resulted image is transformed (a constant value that equals the range of the gray level variation of the background is subtracted) to suppress the black side of histogram, Fig. 2c. So the image, in which the cell structures are visible as the white and gray tiny structures on the black background, is constructed. The image is blurred using the Gaussian filter to mark the region around the structures, Fig. 2d. The threshold (about  $t=1$ ) with the hole filling procedure separates the regions with cells, Fig. 2e and f.

*Regions type discrimination.* To determine the texture, the original image after the background subtraction is transformed by the following operation:

$$q(x, y) = \begin{cases} f(x, y) - f_{med} & \text{for } f(x, y) \geq f_{med} \\ f_{med} - f(x, y) - 1 & \text{for } f(x, y) < f_{med} \end{cases} \quad (1)$$

where  $f_{med}$  is a median of the gray level, Fig. 2h. This artificial image is nearly black, because the mean gray level of the image is moved to black, while all the cell details are preserved by slight darkening of all details brighter than the background and changing to bright all the cell structures darker than the background.



**Fig. 2.** The image (a) and partial results of the proposed method: the background suppression (b, c), the regions detection (d, e), regions mask superimposed on the original image (f), chosen regions and their histograms in the original image (g) and in image constructed for the region classification (h): the converged cells regions (the lowest cell and lowest histograms), the flattened cells regions (the upper right side cell and the upper histograms) and the transitional cells regions (the middle left cells and the middle histograms) marked by ellipses in g and h

The first-order statistics: the mean value of gray level, variance, skewness and kurtosis of the normalized gray level histogram [20] are computed from the histogram of the transformed image in the regions found in the previous step. The three bottom histograms in Fig. 2h show histograms for the regions: in the top row for the region covered by the flattened cells, in the middle one by the transitional cells, while in the bottom one by the converged cells.

The histogram considers the gray level of each pixel separately and no spatial information is conveyed in these parameters. To incorporate the spatial distribution of the gray levels, the region under consideration is divided into texture windows and all features examined in the previous step are examined in each window. The map of the texture is used to examine homogeneity of the region and divide the region into sub-regions, if necessary, see the upper part of Fig. 2g with the white texture window and the black border between homogeneous sub-regions.

The separate method is proposed to count cells in the each covered by the cells region type.

*Converged cells' region counting procedure.* The boundaries of the converged cells with a strong halo and an intensive texture and a certain shift in gray level are visible. So the Sobel gradient is calculated and the threshold is stated for it to show the strongest gradient net. This irregular net is analyzed to find the quantity of meshes in it.

*Flattened cells region counting procedure.* For the flattened cell region it is expected that only one cell is placed in one region. The gray level of the cell cytoplasm around the cell nuclei and the endoplasmatic reticulum is not significantly different from the background, so it is generally included to the background. To avoid the seldom situation when more than one cell is situated in this type of the region the shape of the region is examined if there are necks in these regions to break it. It happens when two cell structures are placed close to the cellular border and close one to the other (Fig. 2e regions marked by \*). Some pseudopodia or bother of the flattened cell are segmented as the separate objects, but almost in all such cases they are excluded from the analysis because of their small size, (Fig. 2e regions marked by !).

*Transitional cells region counting procedure.* The most difficult case is to count the cells in the region covered by the transitional cells. Sobel gradient lines are not complete in these regions. To support an information about the cell border, the map of the probability that the pixel belongs to the nucleus is constructed based on the texture. But in many cases the map conflicts with the gradient, so the area of the whole region is divided according to the area of regular cells of this type to estimate the cell quantity.

## 2.2 Expert to Automatic Counting Result Comparison

The results of expert counting and automatic counting with the proposed method were compared using the image sequences which were collected in the NeuroRepair Department Laboratory, Polish Academy of Sciences Medical Research Center, based on the Human Umbilical Cord Blood derived Neural Stem Cells (HUCB-NSC line), established in this laboratory [21, 22]. Images (2048x2048 pixels) were acquired up to 46 h in every 15 or 20 min in a chosen plane of 120x120  $mm^2$  using the inverted confocal laser scanner microscope with red colour [Zeiss].

After the automatic classification of the 20 randomly chosen regions of each type from the 10 randomly chosen images the results were compared to the visual counting, obtained from the observer blind to the automatic results. Bias of two compared methods were estimated by the mean difference and the standard deviation of the difference:

$$D = \frac{N_A - N_M}{\frac{1}{2}(N_A + N_M)} \quad (2)$$

where  $N_A$  is a number of cells counted by the proposed method and  $N_M$  is a number of cells counted by an operator.

### 3 Results

Classification of the regions were not evaluated, but the quantity of cells was evaluated with the mean difference at two levels: the image level and each type of region separately, to compare the reliability of each counting procedure. So the overall mean difference and its standard deviation were equal  $18 \pm 16,85\%$  for the image, while the same values computed for each region separately according to the quantity of cells were found for: the converged cells  $8,4 \pm 7,05\%$ ; the flattened cells  $11,21 \pm 1,3\%$ ; the transitional cells  $25,8 \pm 17,7\%$ .

### 4 Conclusions

The method has been developed and used to quantitative cell counting in the laser scanning microscopy images of the HUCB-NSC cell cultures. It was found that reliability of the method increased with the increase of the cell quantity and that the automatic procedures were useful. Preliminary results showed that when a large quantity of the cells in one image was counted, than the accuracy varied according to the cell type, but the global accuracy was similar to that provided by an expert. The next steps of the method improvement were - increasing the accuracy in estimation of quantity of the transitional cells by including information about the temporary cell position (the previous frame or even frames) and - examining the discrimination power of all six features used to the region and window classification.

The method can be adopted to other cell types or microscopic images by adjusting the low level cell counting in the region, according to the new setup or subject. The selection of the texture features of regions, based on the feature discrimination and investigation planned in the next stage of the project should be a tool to adjust the method to the situation.

Changes of cell quantity in time enable verifying the model of cell clones growth. It is needed to validate the cell culture growth in real time, which is the goal of our full project. The system with the cell culture validation is prepared for the study on the influence of various substances, such as growth factors, and conditions on the cell culture development.

### Acknowledgement

I am grateful to the NeuroRepair Department Laboratory, Polish Academy of Sciences Medical Research Center for giving the run of cells from the HUCB-NSC line to us and the support in preparing experimental setup.

### References

1. Boezeman J, Raymakers R, Vierwinden G et al. (1997) *Cytometry* 28:305-131
2. Hoppe A, Wertheim D et al. (1999) *Medical and Biological Engineering and Computing* 37(4):419-423
3. Korzynska A (2002) *Annals of The New York Academy of Sciences* 972:139-143

4. Pietka D, Dulewicz A, Jaszczak P (2005) Removing Artefacts from Microscopic Images of Cytological Smears. A Shape-Based Approach. In: Kurzynski M, et al. (eds) Computer Recognition Systems, Springer-Verlag, Heidelberg
5. Wouwer GVD, Weyn B, Scheunders P et al. (2000) *J. Microsc.* 197:25-35
6. Beil M et al. (1995) *Computer Meth Programs Biomedicine*, 48:211-219
7. Malpica N, Ortiz C, Vaquero JJ, et al. (1997) *Cytometry*, 28:289-297.
8. Korzyska A, Jurga M, Dzwigala M, Domanska-Janiak K, Strojny W (2006) Behaviour of Neural Stem Cells in Culture Investigation Abstract book of Cytokinematics:31
9. Fu KS, Mui JK (1987) *Pattern Recognition* 13(1):3-16
10. Liedtke CE, Gahm T, Kappei F et al. (1987) *Analyt Quant Cytol Histol* 9: 197-211
11. Cocquerez JP et al. (1995) *Analyse d'images: filtrage et segmentation*. Masson
12. Korzyska A, Strojny W, Hoppe A, Wertheim D, Hoser P, (2007) Segmentation of microscope images of living cells, accepted by *Pattern Anal Applic.* In DOI 10.1007/s10044-007-0069-7
13. Bellet F, Salotti JM, Garbay C (1995) *Traitement du Signal* 12(5):479-494
14. Proffitt RT, Tran JV Reynolds CP (1996) *Cytometry*, 24:204-213
15. Zicha D, Dann G (1995) *Journal of Microscopy* 179:11-21
16. Alberts B, Bray D, Lewis J, Raff M, Roberts K, Watson JD (1994) *Molecular Biology of the Cell*, third edition. Garland Publishing Inc., New York, London
17. Korzyska A, Jurga M, Domaska-Janik K, Strojny W, Woskowicz D (2005) Analysis of Stem Cell Clonal Growth, In: Kurzynski M et al. (eds) *Computer Recognition Systems*, Springer-Verlag, Heidelberg
18. Comaniciu D, Meer P (2001) Cell image segmentation for diagnostic pathology. In Suri JS et al. (Eds) *Advanced algorithmic approaches to medical image segmentation: state-of-the-art application in cardiology, neurology, mammography and pathology*, 541-558
19. Gonzalez RC et al. (2001) *Digital Image Processing*. Prentice-Hall, New Jersey
20. Kulikowski JL, Przytulska M and Wierzbicka D (2005) A Method of Supervised Discrimination of Texture Based on Serial Tests. In: Kurzynski M et al. (eds) *Computer Recognition Systems*, Springer-Verlag, Heidelberg
21. Buzanska L, Jurga M et al.(2006) *Stem Cell and Development* 15:391-406
22. Buzanska L, Machaj EK, Zablocka B et al. (2002) *J Cell Sci* 115:2131-213



---

# Value of Digital Image Analysis in Research and Diagnosis of Urine Bladder Cancer

A. Dulewicz, D. Pietka, and P. Jaszczak

Institute of Biocybernetics and Biomedical Engineering PAS, 02-109 Warsaw, ul.  
Ks. Trojdena 4, Poland  
ibib@ibib.waw.pl

**Summary.** In this paper two methods of detecting neoplastic changes in urinary bladder are proposed: a noninvasive method mainly for screening investigations based on image analysis of specimens obtained from urine and quick little invasive method to be used as a support for clinical diagnosis based on material obtained by bladder washing technique. All together 147 cases were analyzed with the help of digital image processing system designed by the authors. Multistage classifiers were constructed to identify positive and negative cases. The results of this study yielded a 71% correct classification rate in the control group, while a 100% rate was obtained among the cancer patients for the noninvasive method. The positive test predictive value was equal to 0,77 and negative test predictive value was equal to 1. For the quick (little invasive method) the results were: a 68% correct classification rate in the control group, while a 86% rate was obtained among the cancer patients. The positive test predictive value was equal to 0,82 and negative test predictive value was equal to 0,73. The results show that both methods could be very promising additional diagnostic tools in detection of urinary bladder cancer.

## 1 Introduction

Urinary bladder cancer belongs to the 9 most frequent cancers occurring in the world [2]. It causes 330000 new cases annually and death of 100000 persons. In Poland it is one of the 12 most common cancers and ranks fourth as a cause of male mortality. That is why there is an urgent need for working out an easy, noninvasive method of detecting neoplastic changes at their early stages and for a quick method of monitoring cancer progression or regression in the course of therapy. Nowadays, the regular examination of patient consists in: collecting of complete medical histories of patients (risk factors and symptoms), physical examination with a view to determine the size of bladder tumor, histological examinations. For histological examination which is crucial patients undergo transurethral resection of the tumor which is an invasive test often causing inflammation, dysuria, hematuria.

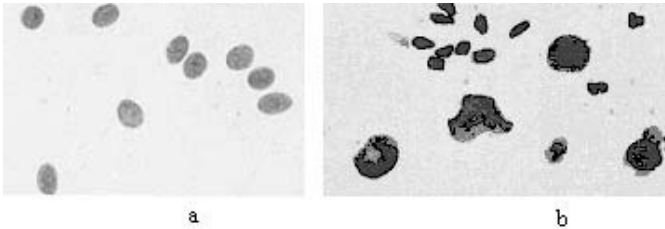
We present two methods: a noninvasive method based on image analysis of specimens obtained from urine and quick little invasive method based on material obtained by bladder washing technique. This technique consists in washing out bladders with physiologic saline.

## 2 Materials and Methods

Urinary bladder tumours originate from lining and glandular epithelium of urinary bladder wall or from other cells of bladder wall. Tumour cells exfoliate into urine or are washed out with physiologic saline during bladder washing. Neoplastic cells differ from eginormal as they have:

- different structure of nuclei,
- increased ratio of nucleus to cytoplasm,
- bigger nuclei than in normal cells,
- irregular shapes of nuclei,
- nuclei contain usually more chromatin,
- nuclei have enlarged nucleoli,
- sometimes bigger number of nucleoli,
- more of them exfoliate.

We concentrated on analysis of nuclei only (not a whole cells), as changes in nuclei structure seemed to be the most characteristic for neoplastic transformations. Examples of normal and neoplastic nuclei are presented in Fig.1.



**Fig. 1.** Examples of normal nuclei (a), examples of neoplastic nuclei (b)

Both presented methods are computer-based analysis of Feulgen-stained urinary bladder cell nuclei.

The first non-invasive method was based on specimens observed under an optical microscope at an objective magnification of 60:1.

In the second method smears wear observed under an optical microscope (10x). The processes of image scanning, processing, analysis and identification of positive and negative cases were performed automatically in both methods by means of our own software.

Image analysis performed in both methods consisted in 4 stages [1], [3], [5]:

- Shade correction – image background correction,
- Image intensity normalization,
- Objects extraction,
- Measurement of selected features.

The last stage of the methods: classification was realized in different way in these two methods as was based on different features in those methods.

The method based on analysis of smears obtained from urine was based on analysis of morphological features. 43 features were defined in order to describe extracted objects [1]:

- geometric features (area, perimeter, coefficient of shape etc.)
- textural statistic features describing inner structure of objects calculated on the basis of light distribution within an object, such as mean value, variance, total optical density etc.
- textural topographic features describing topography of chromatin distribution inside an object, location of center of gravity, moment of inertia.

The method based on analysis of smears obtained by “bladder washing” was based on analysis of objects sizes distribution in population of cell nuclei.

6 features were defined in order to describe extracted objects:

- object size distribution in three ranges of pixels size,
- ratio of the number of object clusters to the global number of single objects.

The specimens were prepared and diagnosed in:

- Department of Histology and Embriology, Centre of Biostructure Research, Medical Academy, Warsaw, Poland.
- Department of Urology at University Hospital of Nijmegen and later by Leiden Pathological Laboratory, Holland.

## 2.1 The Method Based on Analysis of Smears Obtained from Urine

In the method based on analysis of smears obtained from urine 56 patients were examined. 23 of them had no cancer (control), 33 were diagnosed as having bladder cancer of G1,G2,G3 grades of malignancy.

Classification algorithm for this method was based on data from 12676 objects derived from 8 healthy people and 21 patients. Statistical packages: Statgraphics 7.0 and SPSS 10.0 were used for a stepwise selection of features. 39 candidate predictors passed the tolerance test and entered the model. The value for entry was  $\geq 3,84$  and F for removal was 2,71. The calculation of linear discriminant functions resulted in four discriminant functions. Whereas, the final result at this stage of classification gave 53,49% correct reclassifications. Therefore, a model of multistage classifier was chosen [4]. The decision tree showing the algorithm of classification shows Fig.2.

In each node of the decision tree a different decision rule was used. Those decision rules were especially constructed for each node on the basis of a training data specially selected for the particular tasks [3].

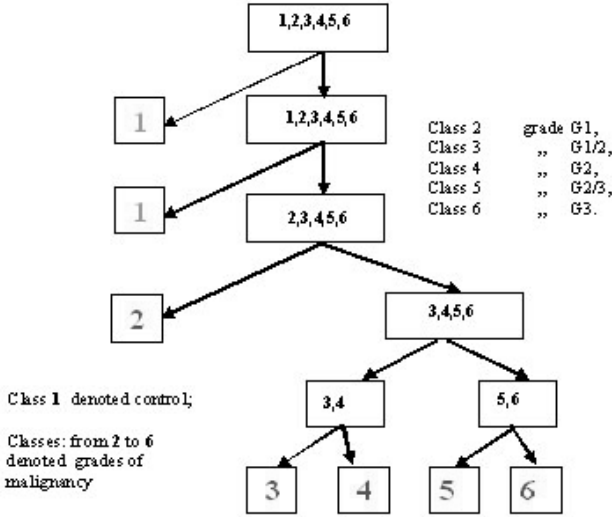


Fig. 2. The decision tree showing the algorithm of classification

Results of case classification for this method on the learning data was:

- 100% correct classifications to the control group,
- 95,2% correct classifications to the group of cancer patients.

Results of case classification for the testing data was:

- 71% correct classifications for the control group (21 new cases),
- 100% correct classifications for the group of cancer patients (21 new cases).

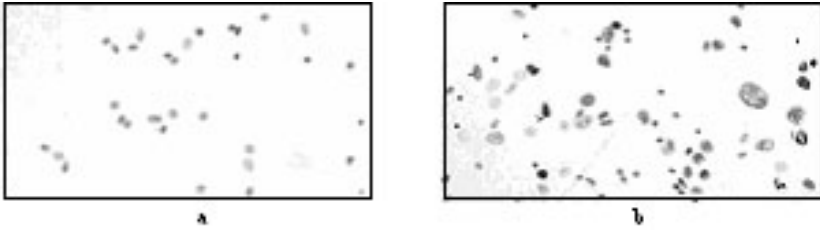
The result of 100% correct classifications for the group of cancer patients was very important because it meant that there was not false negative diagnosis what is very important particularly in screening examinations.

## 2.2 The Method Based on Analysis of Smears Obtained by "Bladder Washing"

In this method 56 patients were examined as a teaching set and 76 as testing set. 48 examined persons had no cancer. They worked as a control group. 39 persons were diagnosed as having bladder cancer of low grade histological malignancy and 45 persons as having cancer of high grade histological malignancy.

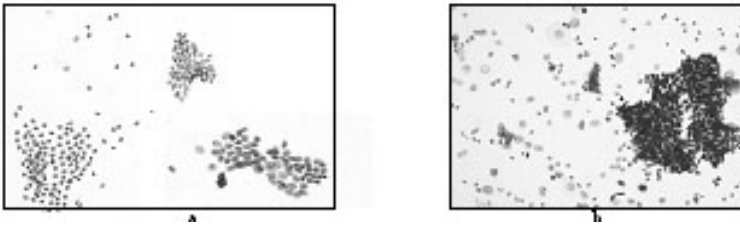
The typical examples of nuclei from the control and high malignancy groups are shown in Fig. 3.

One of the very important feature concerning patients having cancer is greater exfoliation of urothelial cells than with healthy people. That means, that density of nuclei could be a good feature for discriminating between control and malignancy. Another observation is relatively big differences in size of nuclei clusters



**Fig. 3.** The typical examples of nuclei from the control group(a) and high grade malignancy group (b)

and single nuclei with patients cancer and healthy people. The typical examples of nuclei clusters from the normal and high grade malignancy groups are shown in Fig.4.



**Fig. 4.** The typical examples of nuclei clusters from the normal (a) and high grade malignancy group (b)

Analysis of nuclei size histograms of control, low grade and high grade of malignancy showed the biggest differentiation between evaluated groups of cases in the range of 5 – 150 pixels, what corresponds to the range of  $3,6\mu\text{m}^2$  -  $108\mu\text{m}^2$ . In Fig.5 histograms of nuclei size distribution for those three groups are presented.

The greatest difference between control and malignancy was observed in the range 5 – 40 pixels ( $3,6-29 \mu\text{m}^2$ ) and in the range 80 – 120 ( $58-86 \mu\text{m}^2$ ). The greatest difference between high grade and low grade was observed in the range 60 – 120 pixels ( $43-86 \mu\text{m}^2$ ).

Analysis of nuclei mass distribution also showed significant differences between analysed groups (Fig.6).

Observed differences of mean value of nuclei number for respective groups were grounds for definition of 6 parameters representing differences between groups. Five of them were defined as ratio of number of objects in five different intervals of objects sizes, other was the measure of the ratio of the number of nuclei clusters to the global number of single nuclei and granulocytes in the smear.

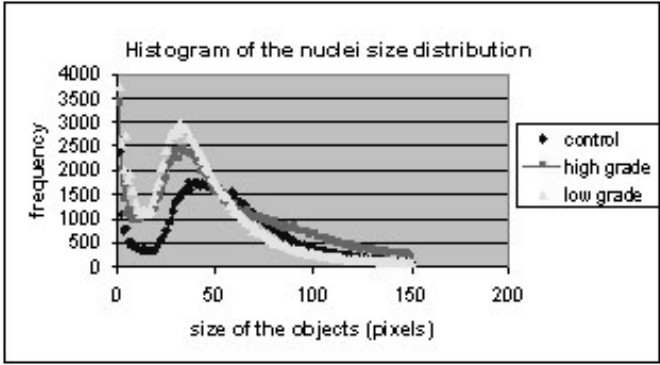


Fig. 5. Histograms of the nuclei size distribution for control, high grade and low grade

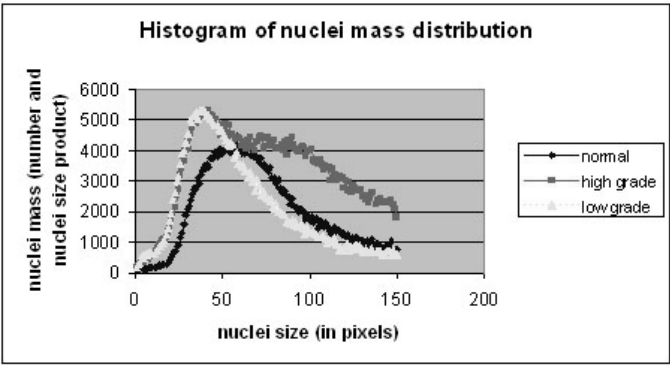


Fig. 6. Histograms of the nuclei mass distribution for control, high grade and low grade

The model of a multistages classifier was chosen. The algorithm of classification shows Fig.7.

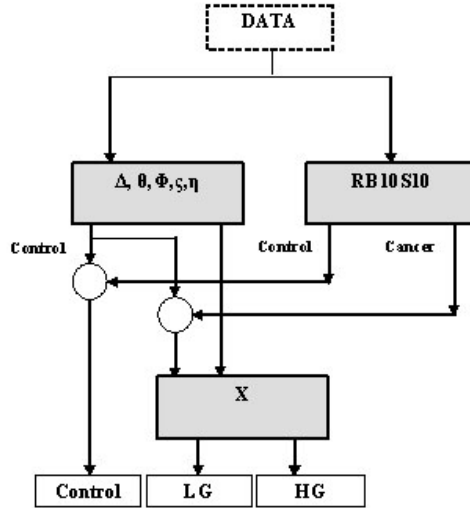
The result of the classification the testing data was:

- 68% correct classifications for control group and
- ~85% correct classifications for malignancy group.

It meant that:

- the false positive ratio was 32%, and
- the false negative ratio was ~15%.

This indicated that the sensitivity of the method was ~68%, and the specificity ~85% what indicated to usefulness of this test for applications in clinical practice.



**Fig. 7.** The algorithm of classification, where:  $\Delta, \theta, \Phi, \varsigma, \eta, X$  – parameters defined as ratio of the number of objects in five different intervals of objects sizes, RB10S10 – ratio of the number of nuclei clusters to the global number of single objects (nuclei and granulocytes) in the smear, Control – denoted norm (no cancer), LG – Low grade of histological malignancy, HG – High grade of histological malignancy

### 3 Results and Conclusions

- The method based on analysis of smears obtained from urine.

**Sensitivity** of this method  $\left(\frac{TP}{TP+FN}\right)$  was  $\sim 71\%$

**Specificity:**  $\left(\frac{TN}{TN+FP}\right) = 100\%$

**Positive test predictive value**  $\left(\frac{TP}{TP+FP}\right) = 0,77$

**Negative test predictive value**  $\left(\frac{TN}{TN+FN}\right) = 1$

**Accuracy** of the method  $\left(\frac{TP+TN}{TP+FP+TN+FN}\right) = 0,86\%$

where  $TP$  denotes - True Positives,  $FP$  denotes - False Positives,  
 $TN$  denotes - True Negatives,  $FN$  denotes - False Negatives.

- The method based on analysis of smears obtained by bladder washing.

**Sensitivity** of this method  $\left(\frac{TP}{TP+FN}\right)$  was  $\sim 85\%$

**Specificity**  $\left(\frac{TN}{TN+FP}\right) = \sim 68\%$

**Positive test predictive value**  $\left(\frac{TP}{TP+FP}\right) = 0,82$

**Negative test predictive value**  $\left(\frac{TN}{TN+FN}\right) = 0,73$

**Accuracy** of the method  $\left(\frac{TP+TN}{TP+FP+TN+FN}\right) = 0,79\%$

Positive and negative predictive values of tests are very important in diagnostic practice because they show what is the probability that a positive test is true positive and what is the probability that a negative test is true negative.

Both methods can be still improved by adapting them to investigated populations. They should be constructed on the basis of data of the same prevalence as the testing data. Prevalence is the number of infected persons at a given time compared with the total in the tested population. Then predictive values of positive test will increase in examination of population of bigger prevalence and predictive values of negative test will increase in examination of population of lower prevalence.

It seems that computer-aided systems could be very promising additional diagnostic tools in detection of urinary bladder cancer both in screening examinations of high-risk populations (on the basis of urine analysis as non-invasive method) and in clinical practice greatly contributing to a rapid confirmation of the diagnosis made by the physicians.

## References

1. Dulewicz A, Pietka D, Jaszczak P, Nechay A, Sawicki W, Pykao R, Kozmiska E, Borkowski A, (2001) Computer identification of neoplastic urothelial nuclei from the bladder. In *Analytical and Quantitative Cytology and Histology*, vol.23, (5), October, 321-329.
2. Boon M.E, Drijver J.S, (1986). *Routine cytological staining techniques. Theoretical Background and Practice*. Macmillan Education Ltd., London.
3. Dulewicz A, Pietka D, Jaszczak P (2005) Trial of practical computer analysis of urothelial nuclei for cancer detection. In: *Progress in bladder cancer research*, Nova Biomedical Books, New York.
4. Kurzynski M (1997) *Rozpoznawanie obrazw*. Oficyna Wydawnicza Politechniki Wroclawskiej. Wroclaw.
5. Russ J.C (1995) *The Image Processing handbook*. CRC Press, Boca Raton, Ann Arbor, London, Tokyo,



---

# Movement Tracking of Coronary Artery Segment in Angiographic Images Sequences by Template Matching Method

Hanna Goszczyńska

Institute of Biocybernetics and Biomedical Engineering PAS, Trojdena str. 4,  
02-109 Warsaw, Poland  
hania.goszczyńska@ibib.waw.pl

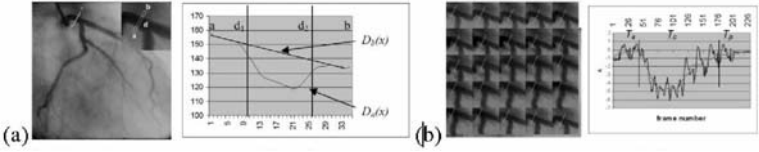
**Summary.** A problem of cyclic movement of measurement field (artery segment) occurs in the approaches focused on coronary flow measurement based on densitometric analysis of coronarographic images. The realized algorithm of automatic artery segment tracking based on template matching method makes possible to trace the results of automatic detection of characteristic points within the structure of arteries, correction of fault matching and incorporation of received movement trajectories for analysis of both the sequence range, where there occurs the lack of X-ray indicator within the tested artery fragment (so its not visible on the screen) and the part of myocardium, close to the characteristic point, what makes possible to the myocardium perfusion estimation on the basis of coronarographic images. Estimated error of the automatic analysis is less than 11%. Introduced system is acceptable for routine clinical testing due to short time for sequence analysis.

## 1 Introduction

The patient motion artifacts, not keeping the breath, organs natural movement cause great problems in performance of different approaches in contemporary medical image diagnostic methods like Digital Subtraction Angiography, registration technique, support of therapeutic intervention or morphological and functional analysis (e.g. flow or perfusion estimation based on the angiographic images sequences). In literature there are reported a lot of possible types of motion artefacts and techniques to avoid and to correct them by means of digital image processing. The main problems are the construction of a 2D geometrical transformation which refers to originally 3D deformation, finding the correspondence between two images using only the grey-level information and efficiently computer application of this correspondence [3]. Finding the correspondence (similarity) between two images is the most important element of mentioned above techniques and have been applied either in static or dynamic approaches.

The present work shows the application of correspondence finding technique to the densitometric analysis performed on the image of moving artery segment in X-ray image sequences of coronary arteries in order to coronary flow  $Q$

estimation by analyzing injected indicator concentration. Goszczyńska et al. [1] proposed to calculation of  $Q$  value using analysis of densitometric curves  $A(t)$  for defined artery cross-sectional line marked on the all images in coronarographic sequence. The measurements of the image grey levels are performed along the same line with its adjacent background (Fig.1).



**Fig. 1.** Densitometric curve  $A(t)$  calculation: a) image brightness calculation along the cross-sectional line on the LCA image, b) densitometric curve for the cross-sectional lines marked in an example of coronarographic image sequence

Figure 1a shows brightness curve  $D_a(x)$  along the cross-sectional line of the left coronary artery (LCA) and the approximated background image intensity distribution curve  $D_b(x)$ . The  $D_a(x)$  and  $D_b(x)$  curves allow calculation of the  $A(t)$  curve proportional to the amount of the indicator present under the measuring window (i.e. segment  $d_1-d_2$ , which is the inner diameter of the artery) at time  $t$ :

$$A(t) = \int_{x=d_1}^{d_2} \ln \left( \frac{D_a(x,t)}{D_b(x,t)} \right) dx \tag{1}$$

where  $x$  – the number of pixel of the cross-sectional line in the digitalized space. Figure 1b presents an example of densitometric curve  $A(t)$  for sequence of coronary images.

Apart to the problems connected with densitometric measurements there occurs an additional problem of cyclic movement of measurement field i.e. artery segment that makes changes both within position and shape.

## 2 Method

Known methods of movement detection or measures within images may be generally splitted into two categories [3][4][5][8]: the optical flow and template matching basing on image similarity measures. In the optical flow two base conditions have to be fulfilled: the brightness of the moving object should be constant in time and the movement of the object between two consecutive frames – relatively small. In the case of arteries both those conditions are not fulfilled for the whole frame sequence [13].

Below there is described the method of automatic movement trajectory indication for the artery segment based on template matching [14] and then setting the cross-sectional line coordinates in each frame of the sequence.

Within the analyzed sequence one can indicate three periods: before contrast appearance  $T_a$ , with contrast filling  $T_c$  and wash-out period  $T_p$  for the chosen fragment of artery (Fig. 1b). The movement analysis process for the indicated part of artery had been splitted into two stages: artery movement detection in period  $T_c$  and fault detections correction.

There has been used the similarity analysis for chosen fragments of original frames (without any preprocessing) with chosen template, i.e. the fragment of artery as shown on the frame where the arteries are well visible (at the end of contrast injection). The sum of difference squares  $C_{SKR}$  has been used ( $R, S$  - the compared fragments of images with area  $m \cdot n$ ):

$$C_{SKR} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (R(i, j) - S(i, j))^2 \tag{2}$$

To find within the frame  $I(t+1)$  the area similar to the given area  $R$  within the frame  $I(t)$ , the area  $S$  within the frame  $I(t+1)$ , having the same coordinates and size, as the area  $R$  in the frame  $I(t)$  is defined, then it is enlarged by values  $k_{max}$  and  $l_{max}$  not less than maximal move of the selected fragment between the two frames (Fig. 2).

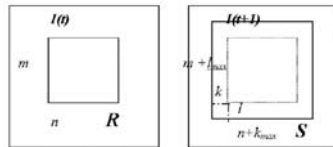


Fig. 2. Determining the template and searching windows

Searching the area within the frame  $I(t+1)$ , most similar to the chosen area  $R$  within the frame  $I(t)$  consists of moving the reference area  $R$  on the magnified area  $S$  and measuring the coefficient of mutual correlation ( $0 \leq k \leq k_{max}$ ,  $0 \leq l \leq l_{max}$ ):

$$C(k, l) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} R(i, j) * S(i + k, j + l)}{\sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} R^2(i, j) * \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} S^2(i + k, j + l)}} \tag{3}$$

The maximal value  $C_{max}$  of this coefficient defines the location of an area the most similar to the reference one. The indexes  $k$  and  $l$ , for which the coefficient takes the maximum, determine the coordinates of the area searched (Fig. 2).

The other similarity measures have been also proposed:

- variance normalized correlation:

$$C = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (R(i, j) - I_R) * (S(i, j) - I_S)}{\sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (R(i, j) - I_R)^2 * \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (S(i, j) - I_S)^2}} \tag{4}$$

where  $I_R, I_S$  – mean brightness values within  $R$  and  $S$  windows, respectively,

- morphological mutual correlation  $CM$ :

$$CM = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \min(R(i, j), S(i, j)) \tag{5}$$

- mutual information coefficient  $C_v$  [6]:

$$C_v = - \sum_{i=0}^I \sum_{j=0}^J P(g_i, g_j) \log \frac{P(g_i, g_j)}{P(g_i) \bullet P(g_j)} \tag{6}$$

where  $P(g_i)$  - the probability of occurrence for the  $i$  grade of grey scale,  $P(g_i, g_j)$  - the probability of occurrence of the pixel with the  $i$ -level gray in one frame and pixel with same coordinates and  $j$ -level gray in the other frame.

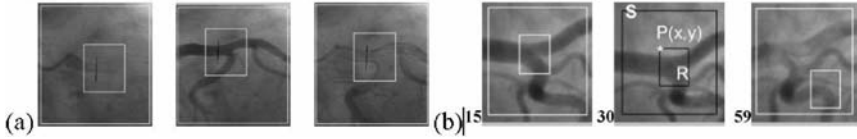
For flow measurement there had been used the manual correction of fault detections during the period  $T_c$  with help of time-related interpolation. To set the movement trajectory in the period  $T_a$  and  $T_p$  there has been used the manual method of time extrapolation.

The matter of operator is setting the frame $_R$  where he sets the window  $R$  containing the template and sets the size  $S$  for the searching area ( Fig. 2). Next the procedure is being started to find the values for indexes  $k, l$  (for each frame in sequence), for which the similarity measure  $C$  of window  $R$  and subarea  $S$  gets maximal value. Then the operator sets the position of cross-section line  $l$  with the length of  $k$  pixels within the frame $_R$  and the procedure of setting the coordinates of cross-sectional lines on  $N$  frames in sequence is started according to the index table received from previous stage. The next stage is the automatic brightness data collection along the preset lines.

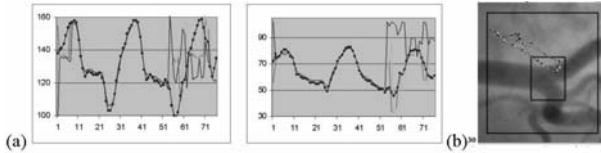
### 3 Results

The Fig. 1b and Fig. 3a show examples of proper matching for all frames within the  $T_c$  range. There have been single fault matching within the analyzed in [1][2] material, like the ones shown in Fig. 5 – mismatching of template  $R$  within window  $S$  of frame 59.

For this particular sequence there have been made the measurements aiming to compare the results of template matching with other similarity measures. Fig.  $P(x, y)$  (Fig. 3b), calculated for the following measures: mutual correlation



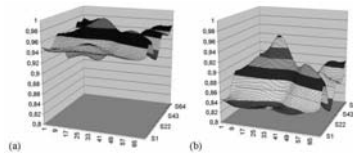
**Fig. 3.** Automatic detection of the cross-sectional line: a) proper matching, b) fault matching (image 30 in image 59)



**Fig. 4.** Point  $P(x,y)$  (Fig. 3b) movement calculation: a) curves  $x(t)$  (a) and  $y(t)$  (b) for sequence of 82 images for three measures and for manual setting (dot curve), b) trajectory of point  $P(x,y)$  for one heart cycle puts on the frame 30

(Eq. 3), normalized correlation (Eq. 4) and morphologic correlation (Eq. 5). The dot curve shows the coordinates of point  $P$  set manually.

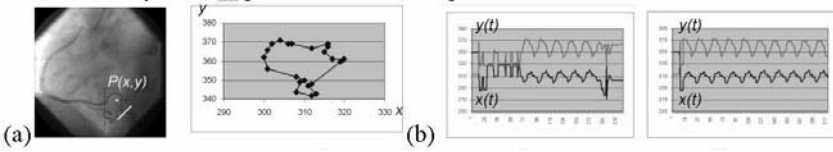
Fig. 4b shows the movement trajectories for the one heart cycle (frames 7 – 35) for three measures and for manual setting (dot curve). The Fig. 5 shows the measures values (3D distance maps) for template matching (the  $R$  window) of the frame 30 to the searching area  $S$  of frame 15 (comp. Fig. 3b) for mutual and morphological correlations.



**Fig. 5.** 3D distance maps for template matching (the  $R$  window) of the frame 30 to the searching area  $S$  of frame 15 for mutual (a) and morphological (b) correlations

During the tests described in [1], in the case of fault matching within the  $T_c$  period (concerning single frames only) the correction has been made by replacing the densitometric data i.e.  $A(t)$  values, obtained from the nearest frames with proper matching.

The Fig. 6a shows the frame of image sequence of the RCA and movement trajectory of point  $P(x,y)$  for one heart cycle. Fig. 6 shows the curves  $x(t)$ ,  $y(t)$  for point  $P$  for all frames before and after time extrapolation by using the



**Fig. 6.** Data movement extrapolation: a) trajectory for one heart cycle for  $P(x,y)$ , b) curves  $x(t)$ ,  $y(t)$  for point  $P$  before and after time extrapolation in  $T_a$  i  $T_p$  periods

repetition of the data of the last heart cycle of the  $T_c$  period within the  $T_p$  period and the first heart cycle of  $T_c$  period within the  $T_a$  period.

### 4 Discussion

The main problems of the realized method for automatic tracing of the artery segment by template matching are the fault matching and time consumption. The main reason for mismatching was choosing of too small searching area  $S$ . The proper setting of the search area has to be based on analysis of maximal movements for chosen artery fragment ( $R$  window) within the  $S$  area. Such fault is easiest to avoid during the analysis and by choosing the bigger searching area (one has to take care, that it would not contain more objects similar to the one wanted for detection) nearly always avoided, but with more time consumption for analysis, that is getting the  $C_{max}$ .

It has been also tested the algorithm, concerning the change of the template during the sequence searching, where template is the result of the former searching. Such algorithm, intuitively better, contains a trap – due to a single mismatching, the template changes totally.

Aside to the problem of finding the best template (templates), choosing the optimal window  $R$  size ( $m, n$ ) and the optimal size of the searching area  $k_{max}$  and  $l_{max}$  the detection faults may also be made lower by taking stronger similarity criteria.

Analyzing the diagrams in Fig. 4a, one is not able to specify the advantages of one algorithm comparing with the other within matching precision, although the morphological correlation has sharper matching peak than the linear correlation (Fig. 5). Basing on the visual mark for diagrams on Fig. 4a, the lowest errors are present for the normalized correlation coefficient.

It is advisable to test and compare the results obtained with the presented method with the methods based on another criterias, as for example usage of 2 templates or incorporation of other similarity measures, i.e. the measure based on histogram analysis for the difference of two adjusted frames [6].

Another problem is a lack of possibility to use the algorithm shown for analysis of fragments of the object, as for example the artery, with no characteristic morphological details, as bifurcations or curvatures. The solution here would be the incorporation of 3D interpolation methods, often used for segmentation or

skeletonization of the coronary arteries, incorporated for frame registration problems with different framing technologies or a 3D image vein image reconstruction in multiplanar registration [15].

Within the problem of fault matching correction another solution intuitively more suitable is a 2D interpolation for the template positioning, basing on the neighbour proper matchings. The too small (especially for the systole period) time resolution of particular probes, what may lead to high level of errors. Apart of this, aiming the point for which the automatic template matching is made, the incorporated solution (of replacing the densitometric data) is not only much easier, but (due to the continuity of the process of contrast medium changing) leads to a lower error level.

The proper matching during the  $T_p$  period is essential on case of perfusion estimation for the part of myocardium around the indicated artery, which movement is estimated on the basis of the closest artery. The changes in density of a part of myocardium are observed within the  $T_c$  and  $T_p$  periods of the artery. The example of prognostic of a single heart cycle for the next one is shown on Fig. 6b. The trajectory analysis may be also used for estimation of the changes in duration of heart cycle phases and as the basis for modeling the epicardial strains [12].

For estimation of an error due to the automatic data collection it has been analyzed the suitability of choice for the following values: the coordinates for cross-sectional line end points, the value of the field under the  $D_a(x)$  curve (Fig. 1a), the value of the field under  $A(t)$  curve (Fig. 1b). As the most suitable the value of area under densitometric curve was chosen. Error caused by the automatic data collection has been estimated at less than 11% [2].

## 5 Conclusion

It seems that the relatively simple algorithm is acceptable for routine clinical testing due to short time for frame sequence analysis (few minutes).

Aside to the occurrence of mismatching, the realized algorithm allowed to automatically collect the data used for testing the blood flow measurement method within the coronary arteries basing on the analysis of diluted indicator. The indicated error of the method doesn't exceed 11%. The realized method of automatic tracing the measurement area not only reduces the manual effort of the operator within positioning the cross-sectional line in the frame sequence, but also allows an easy obtaining of densitometric curve with time resolution equal to the resolution of the tested sequence. The realized algorithm allows also to trace the results of automatic detection of some characteristic points within the arteries structure, correction of mismatching and incorporation of obtained movement trajectories to analyze the sequence range, where there is no contrast within the tested artery segment and the part of surrounding myocardium.

One can list the other solutions of the problem: movement estimation on the basis of phase difference within the Fourier space [10], testing of the effectiveness of incorporation the binary frame after skeletonization as template frame,

incorporation of specialized signal processors [9] or inclusion of the searching strategy into the matching algorithm [7].

## References

1. Goszczyńska H, Kowalczyk L, M. Rewicki (2006) Clinical study on the coronary flow measurements method based on the coronarographic images. *Biocybernetics and Bioengineering*, 26:63-73.
2. Goszczyńska H, Podsiadły T (2006) Flow rate calculation method based on coronarographic images: method error estimation. *Springer IFMBE Proc.* 14: 1-4.
3. Meijering E (2000) Image Enhancement in digital X-ray angiography, PhD Thesis, University Medical Center, Utrecht
4. Coatrieux J-L et al (1995) 2D et 3D Motion Analysis in Digital Subtraction Angiography, *Computer Vision, Virtual Reality and Robotics in Medicine*, (Ayache N. ed.), Springer, CVRMed
5. Konrad J (2000) Motion Detection and Estimation,. In Al Bovik (ed) *Handbook of Image&Video Processing*. Academic Press, San Diego
6. Buzug TM, Weese J, Strasters KC (1998) Motion detection and motion compensation for digital subtraction angiography image enhancement, *Philips J. Res.* 51:203-229.
7. Orkisz M et al (1999) Real-time target tracking applied to improve fragmentation of renal stones in extra-corporeal lithotripsy *Machine Vision and Applications* 11:138-144.
8. Benayoun S, Ayache N, (1998) Dense Non-Rigid Motion Estimation In Sequence of Medical Images Using Differential Constraints", *Int. Journal of Computer Vision* Kluwer Academic Publishers 26:25-40.
9. Bogorodzki P, Wolak T, Pitkowski A (2000) Real Time Implementation of the Parametric Imaging Correlation Algorithms", 59<sup>th</sup> MCB Seminar on Recent Development in MR Imaging, Warsaw
10. Fei R, Gui L, Merzkirch W (1999) Comparative study of correlation-based PIV evaluation methods, *Machine Graphics&Vision* 8:572-578.
11. Meijering EHW, Nissen WJ and Viergever MA (1999) Retrospective motion correction in digital subtraction angiography: a review *IEEE Trans. Med. Imaging*, 18:2-21.
12. Young AA, Hunter PJ, Smaill BH (1992) Estimation of epicardial strain using the motions of coronary bifurcations in biplane cineangiography", *IEEE Trans. Biomed. Eng.*, 39 (5), pp. 526-531.
13. Meunier J et al (1989) Analysis of cardiac motion from coronary cineangiograms: Velocity field computation and decomposition *SPIE* 1090
14. Rong JH (1989) Estimation et caracterisation du mouvement en coronarographie. PhD Thesis, l'Universite de Rennes I
15. Schrijver M (2002) *Angiographic Image Analysis to Assess the Severity of Coronary Stenoses*, Twente University Press, Enschede



---

# Computer Ultrasonic Imaging of the Tongue Shape Changes in the Process of Articulation of Vowels

Krzysztof J. Opielinski, Tadeusz Gudra, and Joachim Migda

Institute of Telecommunications, Teleinformatics and Acoustics  
{krzysztof.opielinski,tadeusz.gudra}@pwr.wroc.pl

**Summary.** This paper presents especially elaborated laboratory setup and computer software for visualizing the tongue surface movement and the changes of its shape using an ultrasonograph (USG). In the paper the acoustic properties of the tongue, the way of examination and the measurement set-up was described, as well. The elaborated setup allows us to computer-visualize and follow in real time the changes of the tongue shape in the process of vowel articulation, thus providing a visual feedback with the speaker. It can be one of the ways of treating speech defects and phoniatric rehabilitation in people with hearing disorders and pronunciation defects.

## 1 Introduction

Ultrasonic examinations (USG) render possible the imaging of the outline of the tongue surface and its muscular structure both while resting and while moving *in vivo* [1, 2, 3]. The purpose of the examination is defining the specifics of the tongue shape and its changes while articulating definite speech vowels, and this can be one of the ways of treating speech defects and phoniatric rehabilitation in people with hearing disorders and pronunciation defects. The essence of this method is the visual feedback between a person subjected to treatment and a tongue shape pattern [1]. It should be emphasized that ultrasonic ultrasonography is non-invasive, painless and harmless for the human body.

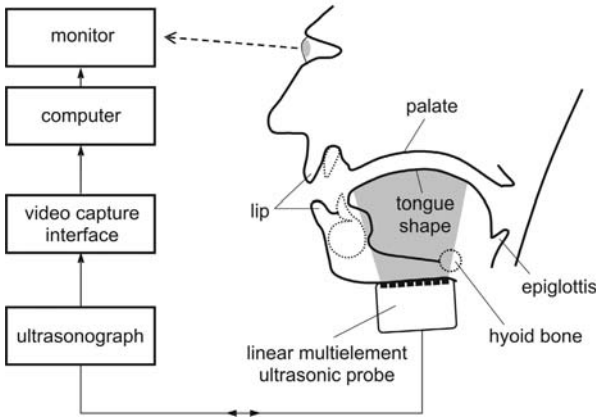
In recent years the idea of transferring images from USG to a computer screen was born [4]. Researchers are in accord on the efficiency of combining the aural and visual feedback in the pronunciation learning process. Computer software can provide the learner with information about the features of the pronounced phrase in real time. By pointing out the features that must be changed the visual imaging provides objective measurements by which both teachers and learners can estimate the correctness of pronunciation. To help in the learning of correct pronunciation of foreign languages what is most often used are computer visualizations of the features of the analyzed speech sound (pitch, frequency, and tempo) [5].

The purpose of this paper was to elaborate laboratory setup and computer software for visualizing the tongue surface movement and the changes of its

shape using an ultrasonograph and with the possibility of comparing the recorded images with a database of appropriate patterns.

## 2 Measurement Setup

The elaborated measurement setup is shown in Fig.1.



**Fig. 1.** Measurement setup for computer ultrasonic visualization of tongue shape changes during speech vowels articulation

The measurement is performed by means of multi-element linear USG probes with working frequency 3.5MHz or 5MHz connected with a Hitachi-Picker LS 2400 ultrasonograph. An image from the video output of the ultrasonograph is captured by means of the Aver TV USB 2.0 interface and sent to the computer through the USB connector.

The most important element of the measurement setup (Fig.1) is the ultrasonograph. Ultrasonography is one of diagnostic methods often used in medicine for imaging organs and tissues of the human body by means of an ultrasonic wave that, propagating in such a medium is subject to, among others, reflection from discontinuity boundaries. The image on the ultrasonograph screen is obtained by means of a set of ultrasonic transducers constituting a multi-element transmitting-receiving probe sending and receiving ultrasonic pulses. On the USG screen what is visualized are amplitudes of returning reflected pulses showing qualitatively the cross-section of the examined organ in the gray scale [6, 7]. What is the most appropriate for examining the tongue shape outline changes is the use of small portable devices. The range of the USG examination should be matched with the tongue size. In typical ultrasonographs setting the range to be 7 cm allows us to obtain 30 frames per second whereas 1.5 cm of range limits their number to 26. The area of optimal spatial resolution of the image is the zone of the ultrasonic beam focusing; a few such zones can usually be set

up, but this lengthens the examination time. One zone placed on the tongue upper surface gives the best resolution [4]. Ultrasonographs are capable of range regulation of amplification in order to improve the brightness of more remote echoes; in the area of the tongue surface the echo amplitude should be increased and in the remaining areas it should be decreased.

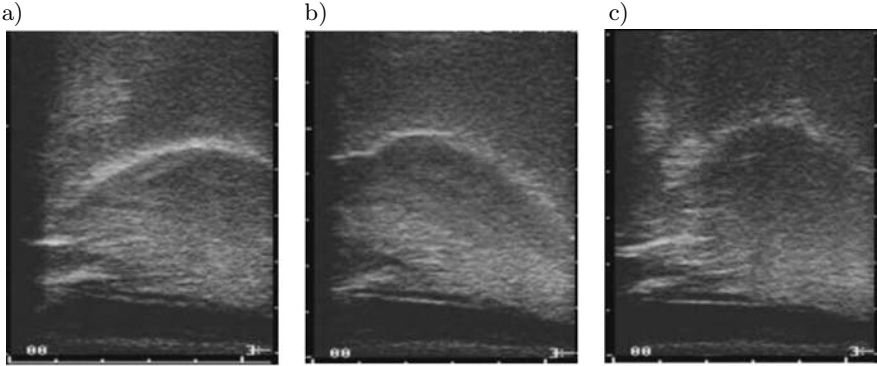
The AverTV USB 2.0 interface manufactured by Avermedia was used to capture the ultrasonograph-obtained image and display it on the computer screen in real time with a possibility of recording and editing. The device has the maximum resolution of 720 x 480 pixels in NTSC standard and 720 x 576 pixels in the PAL standard. It can freeze the image, record both video sequences and single frames, it is easy to install and to operate and is small (PALM type).

In order to minimize the size of the measurement setup a NoteBook laptop can be used.

### 3 Acoustic Properties of the Tongue

The tongue (*lingua* in Latin) is a muscular organ covered with mucous that, with the jaws closed, fills almost completely the oral cavity proper. It is a largely movable organ due to which it plays an important role both in speech and while in chewing and swallowing [8]. The ultrasonic image of the tongue consists of spots visualized in the gray scale and forming edges (Fig.2). The brightest white line is made of echoes reflected from the air just above the tongue surface. What is visualized the most precisely are the edges perpendicular to the ultrasonic beam because they reflect the totality of the beam rays in the direction of the transmitting-receiving probe, whereas oblique or parallel edges direct the reflected beam beyond the probe. The edges of soft tissues are uneven and therefore they reflect the ultrasonic wave in many directions. Most tongue edges reflect the ultrasonic wave well enough to obtain measurable amplitude of the returning echo. And yet the tongue has many features diminishing the ultrasonic wave propagation. The tongue muscles have a digitate structure [8] forming mini-surfaces scattering the sound in the direction from the beam axis both in the penetrating pulse and the returning pulse. Sound dissipation is caused also by the fat included in the tongue volume. Humidity influences the improvement of the ultrasonic wave propagation and the quality of the visualized surfaces. Since the tongue has an irregular surface, when it is covered with saliva it becomes smoother and better reflects the beam.

Different tongues are visualized by means of USG with different quality. Light and lean tongues are better visualized because they contain less fat. Tongues of young people are visualized better than the tongues of elderly people also because of the fat content. Moreover, there is more humidity in the mouths of young people. The best quality of images is obtained for flat and delicately curved tongue surfaces. Because of the ultrasonographs' imaging resolution in time that is 30Hz (30 frames per second) fast movements of the tongue, such as stops, clicks and slight hits are not completely visualized and they should be



**Fig. 2.** Ultrasonic image of longitudinal cross-section of tongue obtained while articulating vowel: a) "a", b) "e", c) "u"; on the left shadow of teeth is visible, on the right - shadow of hyoid bone [3]

measured a number of times so as to get better results by averaging or adding outlines [4].

The tissue-air surface above the tongue reflects over 99% of the wave falling upon it, and that is why the palate is usually invisible in USG images. A similar result in the longitudinal cross-section image is caused by the mandible and the hyoid bone that dissipate the ultrasonic wave before it reaches the tongue surface thus interfering the image in the form of acoustic shadow (black area (Fig.2)) [3]. In the case when the tongue adheres to the palate it is possible to visualize the bottom of the nasal cavity. When the shape of the tongue is hollow (has the shape of an open channel), its middle part is situated lower than its lateral parts, and then obtaining a false USG image is possible (longitudinal cross-section) in the shape of visualizing both the side edge and the middle edge. The reason for these artefacts is the finite lateral resolution of the ultrasonic probe - the beam embraces both edges simultaneously.

## 4 Computer Visualization

In order to computer-visualize the tongue shape changes in the process of speech vowels articulation on the basis of ultrasonograms, an application was elaborated in the object software language C# (C sharp). The main options of the software render possible a connection with the device capturing the image from USG and a real-time visualization, downloading files with images and video sequences, downloading patterns with descriptive information, and going over to the filtration and edition option for images or video sequences. The analyzed image (downloaded or captured from the ultrasonograph) can be immediately compared with patterns, filtered and edited. The menu enables the recording of static images (single frames) and the recording, playback, rewinding and stopping of video sequences. In the database of patterns images and video sequences

recorded for a few speakers pronouncing different speech vowels are stored [3]. The following information is stored with the pattern recorded in a graphic or video file: pronounced speech vowel or word, the data of the user who prepared the pattern, the pattern production date, the pattern production ultrasonograph type, the imaging USG probe type, the probe placement (longitudinal or transverse).

The image filtration option allows us to select one of three filters and set up its operation threshold pointing the starting level of image filtration. Sharpening the outlines and manual graphic edition are also possible. In the case of video sequences, image sharpening and filtration are done with the same settings for all frames. In this case the image edition is not available. In the elaborated software what was chosen and implemented were the algorithms of three different image filters: the nine-pixel filter, the simple gradient and the Roberts gradient [9].

The essence of the nine-pixel filter is a square window measuring 3 x 3 pixels with a distinguished coordinates center  $(x, y)$  in relation to the image filtered. The window is shifted in such a way that each pixel is at a certain step the distinguished pixel. A certain coefficient is ascribed to each pixel of the window in the way shown in Fig.3.

$W_{(-1,-1)}$	$W_{(0,-1)}$	$W_{(1,-1)}$
$W_{(-1,0)}$	$W_{(0,0)}$	$W_{(1,0)}$
$W_{(-1,1)}$	$W_{(0,1)}$	$W_{(1,1)}$

**Fig. 3.** Window coefficient of nine-pixel filter

The sum of all coefficients is 1, and their values are usually assumed to be the same that is  $w_{ij} = 1/9$ . The brightness of the image pixel with coordinates  $(x, y)$  is calculated as the average weighed according to the coefficients  $w_{ij}$ :

$$L'(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j)L(x + i, y + j) \tag{1}$$

The simple gradient uses the differential method of image transformation. For a given function of two variables  $L(x, y)$  the simple gradient is a vector function described by the following equation:

$$\vec{G}(x, y) = \begin{bmatrix} \frac{\partial L}{\partial x} \\ \frac{\partial L}{\partial y} \end{bmatrix} \tag{2}$$

The function gradient marks the direction of the largest change (rise) of the function value. In order to filter the image a modulus of this vector is used, defined by the following equation:

$$|\vec{G}(x, y)| = \sqrt{\left(\frac{\partial L}{\partial x}\right)^2 + \left(\frac{\partial L}{\partial y}\right)^2} \tag{3}$$

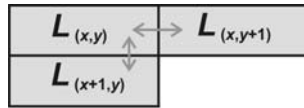
For a digital image, the total differentials can be with good approximation replaced with ordinary differences:

$$|\vec{G}(x, y)| = \sqrt{(L(x, y) - L(x + 1, y))^2 + (L(x, y) - L(x, y + 1))^2} \tag{4}$$

Similar effects are obtained by using the absolute value without squaring and extraction of roots:

$$|\vec{G}(x, y)| \cong |(L(x, y) - L(x + 1, y))| + |(L(x, y) - L(x, y + 1))| \tag{5}$$

Fig.4 shows the principle of calculating the simple gradient.



**Fig. 4.** Principle of calculating simple gradient

Image filtration using the Roberts gradient consists in cross-calculating the difference according to the equation:

$$|\vec{G}(x, y)| = \sqrt{(L(x, y) - L(x + 1, y + 1))^2 + (L(x + 1, y) - L(x, y + 1))^2} \tag{6}$$

or analogically as in the case of the simple gradient:

$$|\vec{G}(x, y)| \cong |(L(x, y) - L(x + 1, y + 1))| + |(L(x + 1, y) - L(x, y + 1))| \tag{7}$$

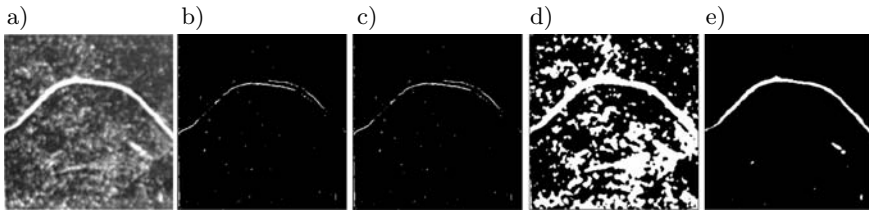
The principle of calculating the Roberts gradient is shown in Fig.5.



**Fig. 5.** Principle of calculating Roberts gradient

Fig.6 shows a comparison of filtration results of an example of USG image of the tongue outline using the simple gradient, the Roberts gradient and the nine-pixel filter.

What results from the comparison is that the simple gradient and the Roberts gradient produce similar results (compare Fig.6b and Fig.6c). Fig.6d shows the image after filtering by means of the nine-pixel filter with a set up low filtration



**Fig. 6.** Ultrasonic tongue outline image while pronouncing correctly vowel "a": a) original, b) after filtering by means of simple gradient, c) after filtering by means of Roberts gradient, d) after filtering by means of nine-pixel filter for low level of filtration, e) after filtering by means of nine-pixel filter for high level of filtration

level, the same as for gradient filters. When a higher level for which the nine-pixel filter produces the best results is set (Fig.6e), gradient filters do not work any more.

A simple edition of the image available in the elaborated application allows us to manually add black or white pixels or areas. In the case then after filtration what is visible apart from the tongue outline are interferences in the shape of white spots, they can be removed this way. Adding white areas enables supplementing the discontinuities in the tongue contour.

The option of comparing images with the pattern allows us to superpose as well as to shift and scale the tongue contour; the image from the ultrasonograph may have a size different than the shape recorded in the database.

## 5 Conclusions

The elaborated laboratory setup allows us to computer-visualize and follow in real time the changes of the tongue shape in the process of vowel articulation, thus providing a visual feedback with the speaker. Depending on the way the probe is set up it is possible to obtain longitudinal cross-sections or transverse cross-sections. Using a presentation of TM type (time motion), it is also possible to image in the speech process selected points of the tongue surface in time.

The best results of the tongue contour extraction from the USG image were obtained by means of the simple nine-pixel filter.

The extension of the software by adding possibilities of recording and visualizing waveforms of a speech sound and its characteristic features is planned.

## References

1. Klajman S, Huber W, Neuman H, Wein B, Bockler R (1988) Ultrasonographische Unterstützung Der Artikulationsanbahnung Bei Gehorlosen Kindern. In: Sprache-Stimme Gehor, 12 (in German)
2. Stone M, Morrish K A, Sonies B C, Shawker T H (1987) Tongue Curvature: a Model of Shape During Vowel Production. In: Folia Phoniatics, 39

3. Gudra T, Opielinski K J, Sobczak T (2006) Ultrasonic Examination of the Tongue Shape Changes in the Process of Articulation of Vowels. In: Archives of Acoustics, 31 (4), suppl
4. Stone M (2005) A Guide to Analysing Tongue Motion from Ultrasound Image. In: Linguistics & Phonetics, 19 (6-7)
5. Lambacher S G (1996) Teaching English Pronunciation Using a Computer Visual Display. In: <http://www.u-aizu.ac.jp/~steeve/york95.html>, University of Aizu
6. Najarian K, Splinter R (2006) Biomedical Signal and Image Processing, CRC Press, USA
7. Fish P (1992) Physics and Instrumentation of Diagnostic Medical Ultrasound, John Wiley & Sons
8. Lasinski W (1970) Anatomia głowy dla stomatologow, Panstwowy Zakad Wydawnictw Lekarskich, Warszawa (in Polish).
9. Jain A K (1989) Fundamentals of Digital Image Processing, Prentice Hall, USA



---

# The Biometric Signals Processing

Zbigniew Gomolka<sup>1</sup> and Tomasz Lewandowski<sup>2</sup>

<sup>1</sup> Rzeszow University, Rejtan Street 16a, 35-310 Rzeszow, Poland  
zgomolka@univ.rzeszow.pl

<sup>2</sup> Institute of Computer Science, State School of Higher Vocational Education,  
Czarniecki Street 16, 37-500 Jaroslaw, Poland  
tomasz\_l@pwszjar.edu.pl

**Summary.** Paper is related to biometrics image compression (finger print and iris images) and refers to the earlier works related to the use of the interferometry images in non touch measurement techniques. The finger prints fringes have similar periodic nature similar to interfered waves, thus we applied a special kind of base functions to perform compression of exemplary images. The second stage of the paper deals with the iris decomposition i.e. a hybrid of Canny Edge detector stage and a Circular Hough Transform to establish iris boundaries and data encryption-compression by Gabor wavelets functions. To ensure correctness of the investigations - comparison to real criminals fingerprint database - searching and recognition case - has been tested. Eventually the comparison of iris templates on exemplary human eye datasets has been presented too.

## 1 Introduction

Image processing in general is used for two somewhat different purposes: improving visual appearance of images to a human viewer, secondly preparing images for measurement of features present[1]. Theoretical foundations of the wavelet transform has been already given by Gabor in 1945, but the practical application started since the end of 80-ties. The example of established standard for Federal Bureau of Investigation - based on the wavelet transform system for collection and compression of criminal's fingerprints is the famous one. Taking into account in particular criminal database resides information about 2 mln persons, i.e. 20 mln images 0.8 MB of uncompressed size each, the efforts to decrease the size of the archives is the corner stone of this issue. Performance of the present police systems regarding compression ratio  $C_R$  is about 20. In the first stage of this paper we are trying to use modified Hsieh wavelet to attain compression ratio  $C_R > 20$ , while still preserving unique personal features in the image, called also as minuties. The clue of the wavelet transform is the assumptions about base function  $\varphi$  also referred to as mother wavelet, which makes us able to realize dilations and translations[2]. For fixed values of  $j$  and  $k$  the mother wavelet may be rewritten as follows:

$$\varphi_{jk}(x) = \text{const} \cdot \varphi(2^j x - k) \quad (1)$$

The *const* value which makes the above equation true equals  $2^{j/2}$ . Two consecutive wavelets from two adjacent scales can be written as a linear convolution of the mother wave and scaling factors  $h(k)$ .

$$\varphi_{j+1,k}(x) = \sum_k h(k)\sqrt{2}\varphi_{jk}(2x - k) \tag{2}$$

where the set of factors  $h(k)$  may be also referred to as a highpass filter.

$$h(k) \stackrel{def}{=} \langle \varphi(x), \sqrt{2}\varphi(2x - k) \rangle \tag{3}$$

By following k-modulation:

$$g(k) = (-1)^k h(1 - k) \tag{4}$$

a set of factors  $g(k)$  i.e. a lowpass filter, may be obtained. The orthonormal basis  $\psi(x)$  is eventually obtained by the use of another convolution of  $g(k)$  with  $\varphi(x)$ :

$$\psi(x) = \sum_k g(k)\sqrt{2}\varphi(2x - k) \tag{5}$$

It is worth noting that the splitting of the image causes the decreasing by half the resolution of the consecutive slides. To solve this problem we used the mutation of this transform. In general, the rule of the growing scale has been transferred to the length of the used filter mask. In the investigations we used two Jun-wei Hsieh wavelets[3] described with coefficient as shown below.

**Table 1.** Coefficients of Hsieh wavelet  $h(k)$

$h(1)$	$h(2)$	$h(3)$	$h(4)$	$h(5)$
1.909686	1.215350	0.344036	0.122197	0.028610

With those assumptions, we did discrete wavelet transform on the set images received from Criminal Laboratory.

## 2 Experiments with Fingerprints

Starting with thresholding method it might be noticed, the  $J$  level signal decomposition (particularly our 2-D images) the number of wavelet coefficients with significant energy is small (due to the relatively large fringes correlation). Thus the losses of energy in a sense of image unique personal details might be represented with smaller number of coefficients. Thus the idea of thresholding wavelet coefficients is a way of cleaning out unimportant details considered to be noise. The relation of signal energy with the square of its norm  $L^2$  and the wavelets

coefficients defines Parseval's theorem[4] which is also valid for the wavelets extending. Upon one dimensional representation of signal with orthonormal base  $\{\Psi_{mn}\}$  this formula becomes:

$$\sum_{m,n \in Z} |(\Psi_{mn}, f)|^2 = \|f\|^2, f \in L^2(R) \tag{6}$$

Although there exists many more or less sophisticated methods of thresholding the two methods can be distinguished i.e. soft and hard eliminations respectively. The thresholding example of signal  $x(t) = t$  with fixed level  $\delta$  we can rewrite in the following formula:

$$y_{mk}(t) = \begin{cases} x(t), & |x(t)| > \delta \\ 0, & |x(t)| \leq \delta \end{cases} \text{ soft thresholding} \tag{7}$$

$$y_{tw}(t) = \begin{cases} \text{sign}(x(t))(|x(t)| - \delta), & |x(t)| > \delta \\ 0, & |x(t)| \leq \delta \end{cases} \text{ hard limited thresholding} \tag{8}$$

With the use of graphic presentation both methods acts with the similar manner (see Fig. 1). From the left side, input signal, hard-thresholding method, in the center results of hard thresholding, left side soft thresholding. The second

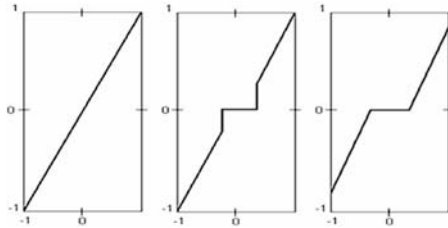


Fig. 1. Threshold mechanism

step while applying this procedure is the choice of a threshold. The consecutive sections briefly discuss chosen method of threshold selection. Donoho and Jonston proposed the estimation of threshold  $\lambda$  based on the formula: Output: if  $x_n$  denotes standard normal random variables, then we can define:

$$A_n = \left\{ \max_{i=1, \dots, n} |x_i| \leq \sqrt{2 \log n} \right\} \tag{9}$$

Also:

$$\pi_n = P(A_n) \rightarrow 0 \text{ when } n \rightarrow \infty \tag{10}$$

In addition, if

$$B_n(t) = \left\{ \max_{i=1, \dots, n} |x_i| > t + \sqrt{2 \log n} \right\} \tag{11}$$

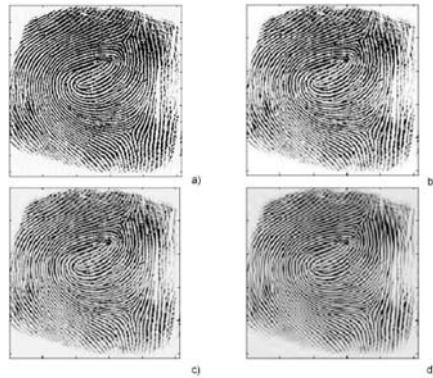
then  $P(B_n(t)) < e^{-\frac{t^2}{2}}$ . Eventually, upon this assumption following universal threshold  $\lambda$  might be established:

$$\lambda = \sqrt{2 \log n} \hat{\sigma} \tag{12}$$

where  $\hat{\sigma}$  the standard estimator of normal noise sequence of a given size.

One may notice, that reduction of coefficients in this method has qualitative rather than quantitative meaning. The all dactyloscopic materials has been kindly facilitated by Criminals Laboratory of Police Headquarters[5]. Taking into considerations the whole images archived by a system for fingerprints acquisition and processing encompasses strictly defined parameters, so we accepted following assumptions for the input images: size: 512x512 pixels, color map: RGB, resolution: 500 dpi, format of data: \*.raw, the capacity in each channel: 8 bit per RGB.

The investigations have been done on the set 50 randomly chosen fingerprints images of anonymous persons. Each file we coded with discrete wavelet transform using 5-type Hsieh wavelet and hard thresholding method, assuming compression ratio coefficient  $C_R$  at rates x20, x33, x100. After that procedure we performed inverse transformation. Resulting images has been shown below. All images included in the test, even with the highest compression ratio were successfully detected/recognized with the use of AFIS system (almost all minuties localized properly, i.e. the probability fingerprint belong to its owner greater than 95%).



**Fig. 2.** Original image a), for  $C_R = 20$  b), for  $C_R = 33$  c), for  $C_R = 100$  d)

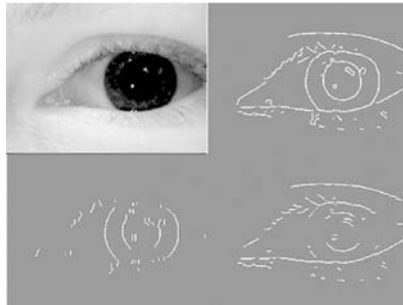
### 3 Iris Data Preprocessing

Although there are already several examples of elaborated working systems with iris recognition the extraction of biometric individuals persists corner stone of this kind data processing. The only most discriminating information present in an iris must be extracted[6], which in fact denotes a kind of data mining and

usually its encoding-compression. Basically the iris is the most visible part of the human eye and its recognition consists of several main steps: image acquisition, image preprocessing, iris localizing, building template, iris matching. An important and most difficult step of an iris recognition system is image acquisition. Since iris is small in size and differs in colors, it is difficult to acquire good images for analysis using the standard devices. Taking this into consideration at this stage of investigation we have already used digitized data from two databases MMU Iris Database and Sabanci University from Istanbul (it seems that lack of iris data for algorithm testing is a main obstacle to research on iris recognition). Each person has a unique iris, moreover the difference even exists between identical "one egg twins" and between their left and right eye. Performing the early first stage of iris pattern matching we are supposed to detect the part of the image that extends from inside the limbus (the border between the sclera and the iris) to the outside of the pupil. We start by determining the outer edge by employing Canny edge detection to generate an edge map. Smoothing module has been attained by formula:

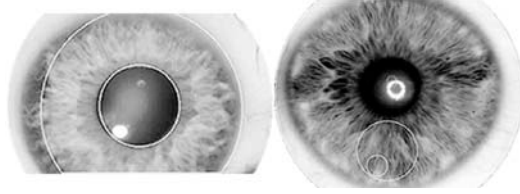
$$g_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (13)$$

where  $\sigma$  denotes standard deviation,  $x, y$  – kernel coordinates respectively. Gradients were biased in the vertical direction for the outer iris-sclera boundary, as proposed by Wildes. The edge operator has been set to the default threshold value given by Matlab environment. The exemplary results have been given at the Fig. 3.



**Fig. 3.** Edge relief map obtained with Canny kernel, input image, a) both directions, b)- vertical and c)- horizontal respectively

With this map of edges the procedure of Hough transform - a circular version has been performed. Two examples depicted on the Fig. 4 represent proper and improper circles extraction. At this stage of computations we supposed there is a very tight correlation between the technical quality and acquisition conditions of the input images (pupil dilation, varying levels of illumination) with correctness of the algorithm (pointers to the next stage of investigation). The tolerance of



**Fig. 4.** The examples of proper and improper results obtained from the Hough module

the radius required to preserve enough small Hough space - i.e. computational complexity acts contrary to the flexibility of a program.

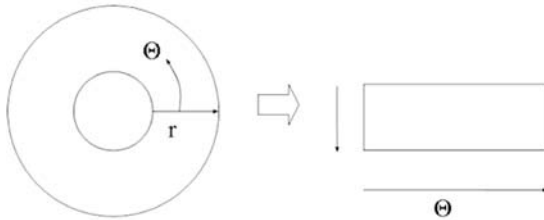
At the next part of computations we applied the homogenous rubber sheet model proposed by Daugman which is also referred to as Log-polar transform - procedure used for object enrollment - called also the contour detection procedure. The model translates all iris points to adjacent pairs of polar coordinates  $(r, \theta)$ , where  $r$  belongs to the interval  $[0,1]$  and  $\theta$  is angle in range  $[0,2\pi]$  see Fig. 5. Generally the transition from Cartesian coordinates to non-concentric area is denoted as:

$$I(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta) \tag{14}$$

where:  $x(r, \theta)$  and  $y(r, \theta)$  are the coordinates corresponding to the  $p$ -pupil and  $s$ -iris boundaries respectively:

$$x(r, \theta) = (1 - r) x_p(\theta) + r x_s(\theta) \tag{15}$$

$$y(r, \theta) = (1 - r) y_p(\theta) + r y_s(\theta) \tag{16}$$



**Fig. 5.** Transition: Cartesians plane  $\rightarrow$  polar plane

Although in this type of the homogeneous rubber model, the size of the pupil might be compensated by the uniformly discretised radius there is still weakness caused by rotational inconsistency (head shifts and rotation). In case of irregular objects, the angle independence might be achieved by computation of its minimum moment axis - unfortunately it can not be applied with the iris (the infinitive number of symmetry axis - perhaps it might be bypassed with the line fixed on two points of eyelids junction).

### 4 Building Template and Iris Matching

In order to provide system with the good accuracy of recognition, the most discriminating information presented in an iris pattern must be extracted. Because of the different circumstances of the consecutive data acquisitions process, only the most significant features of the iris have to be encoded-compressed. That is why Daugman used "demodulation process" of log-Gabor filter to attain the most significant information from an iris. In polar form it might be written:

$$H(r, \Theta) = e^{i\omega(\Theta-\Theta_0)} e^{-(r-r_0)^2/\alpha^2} e^{i\omega(\Theta-\Theta_0)^2/\beta^2} \tag{17}$$

where  $\alpha, \beta$  specifies its width and length, with bandwidth independent mutation:

$$G(f) = \exp\left(\frac{-(\log(f/f_0))^2}{2(\log(\sigma/f_0))^2}\right) \tag{18}$$

where  $f_0$  denotes middle frequency, and  $\sigma$  gives the band width of the filter. We applied the simple measurement of two binary vector location based on the Hamming Distance:

$$HD = \frac{1}{X_j' \cap Y_j'} \sum_{j=1}^N X_j \otimes Y_j \cap X_j' \cap Y_j' \tag{19}$$

where:  $X$  and  $Y$  binary features vectors of adjacent iris images.

Table 2 encompasses  $HD$  distances between four irises and consecutive ten eyes from image database. Assuming the intensity structure of the input images,

**Table 2.** Hamming distance measurement on exemplary four eyes

Nr	hd1	hd2	hd3	hd4
1.	0,4205	0,4205	0,2833	0,1532
2.	0,3644	0,2789	0,1071	0,2674
3.	0,3780	0,4622	0,3939	0,1341
4.	0,3364	0,3788	0,1818	0,3350
5.	0,3235	0,2800	0,1458	0,1364
6.	0,4701	0,4316	0,3438	0,2682
7.	0,2131	0,3889	0,2937	0,3620
8.	0,4444	0,2857	0,3929	0,2891
9.	0,4239	0,3821	0,1087	0,3621
10.	0,3411	0,4092	0,1667	0,2090

real valued (regarding Matlab image containers) i.e. <20x240 double>, the output feature vector is <20x480 1b> each element the compression ratio  $C_R$  can be estimated as:

$$C_R = \frac{IN_{size}}{OUT_{size}} \approx 44 \tag{20}$$

The area pointed by  $IN_{size}$  covers only the polar plane points and do not encompasses the rest pixels of the image.

## 5 Conclusion

The observation made while performing computation with fingerprints and irises are as follow:

- iris patterns possess a high degree of randomness,
- the entropy of the irises images surpasses that of finger prints,
- because of the higher entropy irises might be compressed with higher  $C_R$ ,
- more than double higher compression in iris case ensure better results than processing of fingerprint fringes,
- irises patterns are stable since 7th month of gestation throughout life,

Although the method has good efficiency of humans recognition, there still exist several disadvantages:

- the iris is a small target (1 cm) to acquire from a distance (1 m),
- illumination should not be visible or bright and it could be difficult to fulfill this requirement in real measurement device.

## References

1. Gomolka Z, Kwiatkowski B, Kedzior Z (2003) Improvement of image processing by using neural networks. Mediterranean Conference on Modelling and Simulation, Reggio Calabria - Italy, Full paper CD no.193
2. Kil D, Shin F.B, (1996) Pattern Recognition and Prediction with Applications to Signal Characterization. American Institute of Physics
3. Hsieh J, Liao H.M, Ko M, Fan K (1995) Wavelet-Based Shape from Shading. Graphical Models and Image Processing, vol.57, No.4: 343-362
4. Bialasewicz J.T (2000) Wavelets and approximations. WNT Warsaw
5. Gomolka Z, Kwater T, Twarog B, Pekala R, Bartman J, Kwiatkowski B (2006) Finger print compression with the use of wavelets. VII-th Conference on "Soft Computing", Zakopane
6. Daugman J (2002) Gabor wavelets and statistical pattern recognition. The Handbook of Brain Theory and Neural Networks, 2nd ed., MIT Press: 457-463



---

# Digital Dental Models and 3D Patient Photographs Registration for Orthodontic Documentation and Diagnostic Purposes

Agnieszka Tomaka<sup>1</sup>, Michal Tarnawski<sup>2</sup>, Leszek Luchowski<sup>1</sup>,  
and Barbara Lisniewska-Machorowska<sup>2</sup>

<sup>1</sup> The Institute of Theoretical and Applied Informatics, Bałtycka 5, 44-100  
Gliwice, Poland

{ines,leszek.luchowski}@iitis.gliwice.pl

<sup>2</sup> Silesian Academy of Medicine, pl. Traugutta 2, 44-800 Zabrze, Poland  
michal.tarnawski@ortodonta.com

**Summary.** Orthodontic diagnosis uses the information obtained by different imaging modalities, but routine diagnosis of simple cases still relies on the orthopantomogram, the lateral cephalogram, planar photographs and dental cast models. The article presents a method of virtual registration of 3D photographs (3D facial scans) with digital dental models. By deriving the idea from old gnathostatic models we explain the registration process with the use of registration objects - an equivalent of the traditional facial arch. Potential use of such registration to create a standardized patient dataset is discussed, in order to integrate the analysis of facial and dental models.

## 1 Introduction

Orthodontics is a branch of medicine dealing with the occlusion of the teeth and the function of the masticatory system, and strongly connected with facial aesthetics. To evaluate the geometry of the craniofacial complex, many different images are used. The choice of imaging technique depends on the kind of disease or malformation. A planned surgical intervention is a strong indication to perform CT scans; analysis of location and development of muscles can be obtained using MRI; some cases require bi-orthogonal cephalograms, but routine diagnosis of simple cases still relies: on the lateral cephalogram, planar photographs and dental cast models. This quite considerable amount of image data should be associated together to be exploited in maximum degree and should be properly stored. The integration of the information brought by all these images into a common coordinate system is a major challenge, all the more as new imaging techniques are being introduced [6, 7]. One of such relatively new noninvasive imaging techniques is laser scanning [5], which can be used to obtain 3D patient data. Using visible light, this technique is able to digitize external surfaces of real persons and objects. The present article shows the possibilities of applying the

laser scanning technique to create an integrated standard patient image dataset composed of 3D photograph, digital dental model and lateral cephalogram.

## 2 3D Photograph and Facial Feature Analysis

### 2.1 Traditional Examination

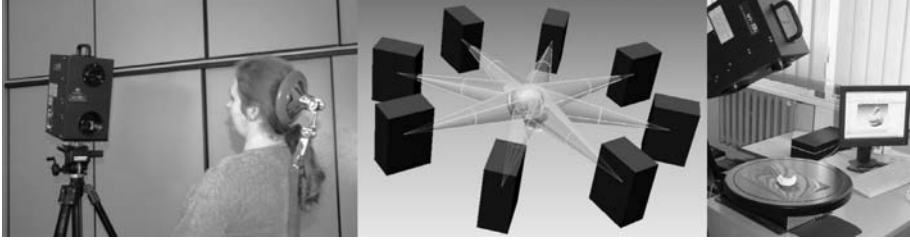
The analysis of facial features is a basic examination that is prerequisite for the differential diagnosis of craniofacial malformation, for the choice of therapeutic management method and for the evaluation of treatment results [8]. Usually, the examination is carried out on the patient positioned in standardized conditions, looking at his face from front and from both sides. Additionally, persons with intensified malformation and asymmetric configuration of facial soft tissues are examined in some local areas, such as the body of the mandible, the base of the nose, or the lip areas.

### 2.2 3D Surface Scanning System

The analysis using photographic method is subject to artefacts and errors resulting from using 2D representations of 3D real structures. The photograph is a projection of 3D structures of the head upon a single plane of the film. Except for a few structures of interest which lie in the plane parallel to the plane of the film, it is impossible to make accurate 2D measurements on photographs, because the structures in planes lying at different distances from camera are enlarged differently. A high-resolution, repeatable surface data acquisition can be obtained scanning the external surfaces by means of laserlight triangulation devices. Laser surface scanning is common in industry and can be used in medicine after assuring the proper conditions for scanning patients[1, 13]. Our 3d head scanning system consists of a KonicaMinolta Vivid 9i 3D scanner moving around the patient and a special chair with a headrest, that prevents breathing-related movement of the patient's head and stabilizes it during the examination. (fig. 1) The scanner can only scan those surfaces which are visible from a given viewpoint. In order to acquire the entire relevant surface of the head it must move around the patient. The scans should be merged together after being brought into register (transformed into a common coordinate system). Most of the processing was done using the RapidForm software environment by InUs Technologies [9]. The determination of scanning parameters and rules for merging scans from different viewpoints are described in detail in [13].

### 2.3 Facial Feature Analysis

The first advantage of using 3D head scans is the possibility to repeat the classic examination procedure at any time, viewing the stored 3D model from any direction (fig. 2), to determine the reference planes and axes of rotation, and then to rotate surfaces by a chosen angular step. Metric analysis of facial asymmetries



**Fig. 1.** Left to right: Scanning a person with the VI-9i; eight viewpoints for an allround image; scanning a dental model



**Fig. 2.** Evaluating facial symmetry. Note how various aspects of the patient's malformity become visible in different positions.

is also made easier by having a digital model, where the differences in position or size of corresponding features on the left and right side of the face can be measured relative to a fixed coordinate system [12, 3].

### 3 Analysis of Dental Models

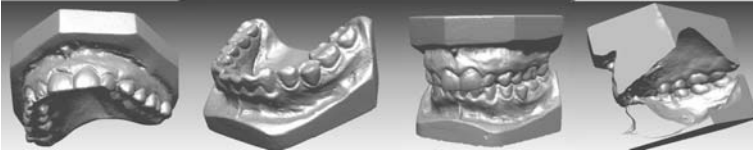
The plaster models used in orthodontics are positive ones, produced by casting plaster into dental impressions. Impressions are taken in the patient's mouth using impression material, which can transit from gel into elastic/solid state. The plaster model allows the practitioner to look on the dentition, palate, gums, and their relation to each other from different positions without tiring the patient. Viewing is useful for descriptive analysis. But clinicians also use metric analyses, measuring such parameters as tooth width, depth of palate vault, and relative position of teeth [8].

#### 3.1 Dental Model Digitization

A number of techniques exist to digitize dental models. One quite popular approach is to physically cut thin slices off the model and take photographs of the successive cross-sections uncovered by each slice. While relatively successful, this method destroys the model and is time consuming.

### 3.2 Our Technique for Scanning the Models and Bringing Them into Register

Like Kuroda [4], we use a 3D laser scanner and a turntable to obtain all-round 3D images of dental plaster models. The models of the upper and lower dental arch are scanned jointly (in occlusal position) as well as separately. The joint scan is not directly relevant to diagnosis, but it allows the separate shapes of the two models to be matched. In practice, we found that the joint scan requires less cleanup processing than the two separate models which will be used for precise measurements. Each separate model is scanned in two positions. It is placed on the rotary table upright (standing on the rear cutoff plane, with the front teeth on top) and flat (palatal or lingual cutoff plane on the table, teeth on top). By using the joint scan as a reference, the two models can be put in register and stored as separate 3D shells whose relative position mimics the patient's occlusion. (fig. 3)



**Fig. 3.** Left to right: scan of upper model, of lower model, joint scan, the two separate scans combined and cut (virtually) to expose internal detail

## 4 Relating the Models to Outer Scans of the Patient's Face and Head

Both 3D photograph and scans of cast models can be acquired quite easily. But the potential for analysis will dramatically increase when the relation between them is known. Two objects can be brought into register when they have some known common parts. Those parts can be the basis for regional registration. The task of the registration is to find a transformation which minimizes the distances between corresponding points (Iterative Closest Point algorithms [10]) or features of both objects. The task however is much more difficult when there is no common region between the objects, as is the case here.

One approach [14] used an X-ray image of both structures (lateral cephalograms) for this purpose. First, they identify the patient's profile on the cephalogram and match the 3D scan of the face to it. Then, based on feature points, they position the dental model relative to the cephalogram. This approach requires a precise knowledge of the projection conditions in the X-ray device, which is not always available.

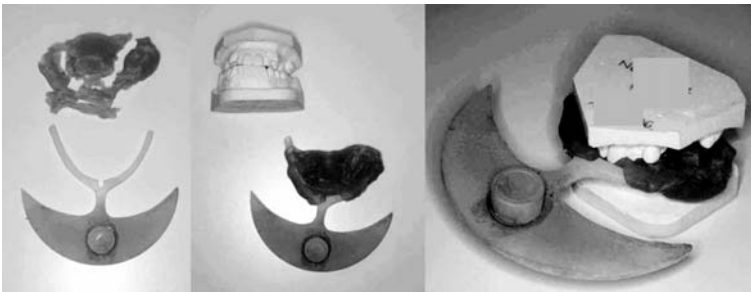
The patient's teeth, in the occlusal position, can be scanned if the patient is asked to grin or if cheek retractors are used to pull the lips and cheeks aside as far as possible. This would yield a 3D image of much of the outer surface of the

teeth as well as the face. However, such scans are not the best way to put the shapes of teeth and of the face in register. Grinning, and especially retracting cheeks mechanically, leads to significant distortion of large areas of the facial soft tissues, and their scanned representation is impossible to match to a 3D image of the same face at rest. Moreover, tooth enamel has a high refractive index and a certain degree of translucency, which makes teeth very difficult to scan reliably.

We propose to use a facial arch to put the 3D images of the plaster models in register with outer facial scans. A facial arch (fig. 4) is a structure that consists of three parts:

- a wax bite (soft malleable body fitting between the patient's teeth without interfering with the occlusion)
- a flat, narrow rigid member protruding outwards between the patient's closed lips;
- a crescent-shaped plate providing a stable geometric reference outside of the patient's mouth.

Simon [8] used similar devices, in the early 20th century, to mechanically reproduce the position of the teeth relative to facial landmarks. This position could then be transferred to a device called an articulator, where the movements at the temporomandibular joints was simulated by a set of gimbals. Alternately, the mechanical bearings (with some additional sliding supports) could be used to position the dental models inside the mouth of a plaster model of the patient's head.

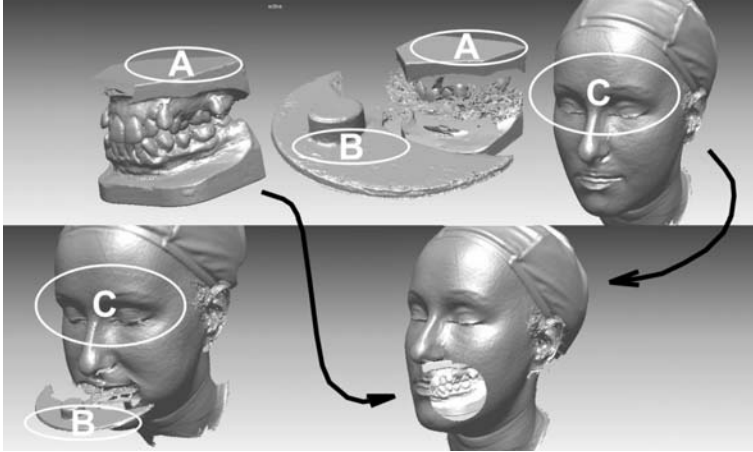


**Fig. 4.** Facial arch. Left to right: arch with wax bite removed; pair of dental models and arch with bite attached; arch inserted between models.

In the present work, we have tested the possibility to achieve a similar reproduction of position (of teeth relative to the face) by digital means. At this stage, we did not attempt to imitate movement.

Therefore to bring into register two structures which do not share such a common part, a third structure can be used which does have a common part with each of them.

By putting the facial arch into the patient's mouth and then - without modifying the shape of the wax bite - between the two plaster models, the plate of



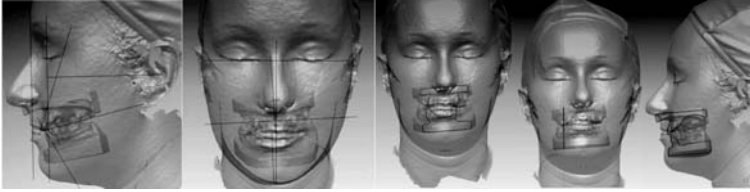
**Fig. 5.** Scan of the two cast models, first without, then with facial arch, a scan of the patient’s face first without, then with the same arch, and the final 3D image of face with the teeth inside and in register. White letters indicate matching areas.

the arch can be used as a common reference structure to put the models and the face in register (fig. 5). The different 3D images are related to each other as follows:

- A–A Scanning the plaster models with the arch inserted identifies the relation between the models (seen as one block) and the crescent of the arch.
- C–C The undistorted region of the eyes and forehead yields the relation between the 3D photograph and the crescent of the arch.
- B–B The positions of the arch can be aligned. The plaster models can be positioned, virtually, inside the undisturbed face. This technique amounts to a digital equivalent of Simon’s mechanical devices.

## 5 Case Study

Our Patient, A.N. had a malocclusion characterized by 3D changes in teeth positioning, deviation of midline, oblique occlusal plane. The analysis of width of upper and lower anterior teeth showed good proportions. The analysis of upper arch geometry revealed serious constriction. The state of the periodontium (gums) didn’t allow the arch to be widened by tipping the teeth. Further analysis was made on cephalograms (X-rays). The position and inclination of the jaws were nearly correct in the lateral projection, but the upper incisors were proclined and the lower retroclined. The postero-anterior cephalogram analysis showed a canting of the maxilla and maxilla displacement towards right. The same results were obtained through analysis of the digitized dental models, and through analysis of digitized models positioned in face scans. The position and



**Fig. 6.** Measurements based on scans of plaster models and face

inclination of palate and canting of occlusion plane were also determined, as well as the inclination of dental crowns in relation to the face (fig. 6).

Obtaining the same results from radiological imagery and by using only laser scans (of face and models) means that now there is a possibility to obtain some data without using the invasive technique of X-rays. It is important because an optical technique can be used frequently and indefinitely, whereas the use of X-ray techniques must be limited. Of course, some data can only be obtained through the use of X-rays, but the possibility of viewing, making crosssection of dentition and face, and manipulating in 3 dimensions, gives the orthodontists a quite new set of data.

## 6 Conclusions

Orthodontists need to acquire and store information about the shape of their patient's faces and dentitions. Porting this information from traditional media like photography and plaster castings to 3D digital model has a number of advantages. The data can be integrated into a common coordinate system, providing the practitioner with a new quality of diagnostic imaging. Using laser scans only (and not X-ray techniques) to bring the various 3D data into register will allow this approach to be used routinely without exposing the patient to unnecessary doses of radiation. The practitioner will also benefit from the usual advantages of digital information processing: ease of storage, access and transmission, versatile presentation, automatic or semiautomatic quantitative evaluation and reporting. Of particular interest is the possibility to compare large sets of data representing many patients and/or many different points in time, to define standards and identify patterns. Future work can reproduce, numerically, another of Simon's inventions, the articulator and model-based kinetic analysis. Mandible kinetics is already being measured directly on the patient by electronic motion capture devices (ultrasound or inductive), but simulating masticatory function on a digital model will help predict the functional impact of treatment.

## Acknowledgement

This work was supported by the Ministry of Science and Higher Education of Polish Government, the research project 3 T11F 004 27.

## References

1. Bianchi S, Spada M et al (2004) Evaluation of scanning parameters for a surface colour laser scanner. *International Congress Series*, 1268:1162–1167
2. Hajeer M, Ayoub A et al (2002) Three-dimensional imaging in orthognathic surgery - the clinical application of a new method, *Int J Adult Orthodon Orthognath Surg* 17:318–330
3. Hajeer M, Ayoub A, Millet D (2002) Three-dimensional assessment of facial soft-tissue asymmetry before and after orthognathic surgery. *British Journal of Oral and Maxillofacial Surgery* 42:396–404
4. Kuroda T, Motohashi N et al (1996) Three-dimensional dental cast analyzing system using laser scanning, *Am J. Orthod. Dentofac. Orthop* 110:365–369
5. Kusnoto B, Evans C (2001) Reliability of a 3D surface laser scanner for orthodontic applications, *Am J. Orthod. Dentofac. Orthop* 122:342–348
6. Mah J, Bumann A, (2001) Technology to Create the Three-Dimensional Patient Record, *Seminars in Orthodontics*, 7:251–257
7. Nagasaka S, Fujimura T, Segoshi K (2003) Development of a non-radiographic cephalometric system, *European Journal of Orthodontics* 25:77–85
8. Rakosi T, Jonas I, Graber T (1993) *Orthodontic Diagnosis*, Georg Thieme Verlag, Thieme Medical Publishers, Stuttgart, NewYork
9. RapidForm 2004 Tutorial, Inus Technology, Inc. 2004
10. Rusinkiewicz Sz, Levoy M, (2001) Efficient Variants of the ICP Algorithm Third International Conference on 3D Digital Imaging and Modeling (3DIM 2001)145–152
11. Terai H, Shimahara M, Sakinaka Y, Tajima S (1999) Accuracy of Integration of Dental Casts in Three-Dimensional Models. *J. Oral Maxillofac. Surgery* 57:662–666
12. Tomaka A, Lisniewska-Machorowska B, (2005) The Application of the 3D Surface Scanning in the Facial Features Analysis *Journal of Medical Infomatics and Technologies* 9:233–240
13. Tomaka A, Luchowski L, Skabek K, Tarnawski M (2005) 3D Head Surface Scanning Techniques For Orthodontics, *Journal of Medical Infomatics and Technologies* 9:123–130
14. Zhao B, Ong S, Foong K (2004) Multimodal Registration of Dental and Facial Images, *Signal and Image Processing SIP* 444:90–107



---

# Gait Motor Disturbances in Neurological Diseases Diagnosis

Jan Piecha<sup>1,2</sup>

<sup>1</sup> Department of Transport Informatics, Silesian University of Technology  
jan.piecha@polsl.pl

<sup>2</sup> Department of Computer Systems, University of Silesia  
piecha@us.edu.pl

**Summary.** The paper introduces some clinical experiments on 250 patients suffering from neurological diseases. Among them: 117 patients with motor disturbances and 42 with hemiparetic syndrome of the disease; majority cases after cerebral stroke. Next 52 cases were affected by acute sciatic neuralgia, 23 patients with recognition of Parkinson disease symptoms. The control group consisted of 16 healthy adults of medical staff. The subjects were examined by pedobarographic equipment; Parotec System for Windows (PSW) [1]. On these observations some recommendations were found; for further development of computer supported diagnostics system.

## 1 Introduction

The gait abnormalities caused by neurological disorders are visible in patients' investigations remarkable different from physiological evidences [2]. The introduced investigations were carried out for several factors of motor disturbances, as:

- posture descriptors and body balance evidences,
- time-range norms with the gait length measures,
- footprint maps on standing (static) and walking (dynamic) cycles.

They were strongly needed for automatic conclusions system currently under development [3], [4]. In neurological diseases clinical diagnosis practices, the traditional investigations are not satisfying enough. Mainly, for specific characteristics, observable in complex disorders, as early Parkinson's disease states. Majority of them can be classified by means of gait characteristics analysis [5], [6]. Till now none of these gait disturbances classifiers have been investigated under electronic equipment implementation. The gait classifiers give the user diagnosis aims, therapy control (rehabilitation and pharmacotherapy [7]) and the disease level judgment.

For example, traditionally applied measures for a stroke level recognition (the Scandinavian Neurological Stroke Scale) are too general for putting any proper diagnosis. They are assigned by five points scale only, as:

- 12 points, for patient able to go through a distance of 5 m; without any help,
- 9 points, for patient able to walk with an equipment support,
- 6 points, for patient supported by somebody,
- 3 points, for patient sitting without any help of the equipment,
- 0 points. for patient laying in bed or sitting in invalid-chair.

The Expanded Disability Status Scale (EDSS) examination also describes some neurological disorders, traditionally used for neurological disease level assignment. It can explain the patient's ability of passing the distance, without stopping and with or without any support (as a stick). These classes are too far from any statement and for medical treatment recommendations.

## 2 Some Motor Disturbances Visible in Neurological Disease

Foot is working as a part of complex mechanism, reacting as shock absorber, of dynamic changes and motor actions, moving the body onto its specific trajectories. In classical diagnostics on gait characteristics analysis, appointed by Mumenthaler's investigations [6], [9] several classes of gait disorders were appointed:

- (a) Paraspastic gait: with a straight shin, trailed feet on a floor with external rotation; visible in multiple sclerosis, cerebrovascular disease and spinal trauma.
- (b) Hypokinetic gait: with short steps, limbs placed slowly and irregularly; visible in Parkinson's disease, cerebrovascular disease, depressions or catatonia.
- (c) Dystonic gait: with irregular and involuntary movement; visible in every step phases of Atetosis, baliom, torsial dystonia and in levodopa.
- (d) Atactic gait: with abnormal co-ordinations, patient strikes limbs that cause disorders of body balance, gait on wide basis ("as sailor"; visible in polyneuropathies, intoxications, cerebellar syndrome; ex. in multiple sclerosis).
- (e) Gait in muscular paresis:
  - Hemiparesis, with an affected limb remarkable overloaded, the gait is hedgehopping, an affected limb is erected in knee, stick is kept at not affected limb side; visible in strokes, after cranio-cerebral injuries, in multiple sclerosis.
  - Isolated, peripheral paresis of foot-muscles rectifier producing "unilateral wading gait"; visible in paralysis of a peroneus nerve in lumbar discopathy; spinal muscular athrophy.
  - Bilateral peripheral paresis of a foot-muscles rectifier producing "bilateral wading gait"; visible in: polyneuropathies, Charcot- Marie- Tooths disease and spinal muscular athrophy.
  - Paresis of knee muscles rectifiers, with flattened affected limb; observed in paralysis of femoral nerve, with bilateral symptoms in myopathies or polymyositis.
  - Paresis of hip adductors: producing disturbances in pelvis's stabilisation, with horizontal position of walking.

In the above neurological diseases, the appointed gait characteristic features are observable allowing gait abnormalities recognition and classification.

### 3 The Motor Disturbances Measure

The above gait disorders list, define the introduced explorations subjects, extracting data from the records gathered by pedobarographic data recorder Parotec System for Windows (PSW) [8], [9]. The neurological diseases are discussed within the same data units.

#### Characteristic Features of Gait Diagrams

The static data distribution of the PSW physiological records corresponds with symmetrical load distribution, i.e. comparable load on both feet.

The gravity centres are located in the middle of a support zone of feet. They show the measures of the body centre dislocation, between both limbs; illustrating the diseases, as:

- a one-sided overload or lateral load,
- an unstable posture disturbing the body balance (postural instability).

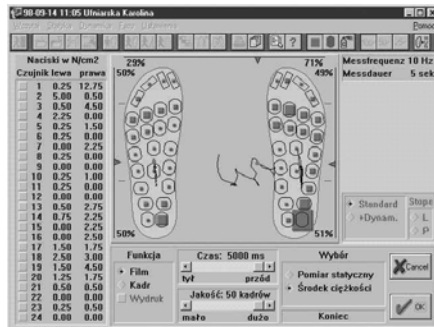


Fig. 1. An example of patient's body balance disturbances

In Fig. 1 an example of remarkable body balance disturbances is presented. The patient's body centre is balancing into his right limb, used for supporting assistance and stabilisation. The body weight centre moves into one side of the body; to the overloaded limb.

Etiology: this characteristic feature was notified for cerebellar lesions, affection of vestibular system, myelosis funicularisa and Parkinson disease.

#### The Data Interfaces

The static data in Fig. 1, describes the body load distribution, where visualisation interfaces provide the user with:

- the body load transfer measure from back to front side of a foot and between left / right limbs (in standing),
- pressure distribution characteristics, for three zones of a foot (heel, front, metatarsus) and a body load movement visualisation, for both limbs (in walking cycle).

In majority cases the static map of a load distribution allows recognise the early state of the disease. The load rotation (body balancing) in standing position suggests neurological reasons of the observed disorders (also Fig. 1).

A static visualisation indicates one-sided load location; at the more affected limb. The pressure load location may also assign the patient's trunk leaning directions; towards suggests the Parkinson's disease symptoms.

In Fig. 2 a large left-sided hemiparesis case, after a brain stroke, is presented.



Fig. 2. A large left-sided hemiparesis after a brain stroke

The 98% of a body weight is carried by the left; an affected limb, with 56% of the load, concentrated within a heel region. The right limb is used for walk assistance only (with 2% of the body weight). The scale of the load transition corresponds strictly with the paresis intensity.

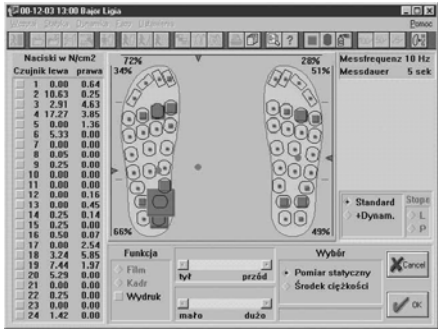


Fig. 3. A case with right-sided Parkinson's disease

In Fig. 3 a static footprint of a right-sided Parkinson’s disease case was introduced. A distinct lateralisation of a body weight and gravity centre, into the left side, was noticed. In majority cases a heel region is burdened, independently from the affected limb.

An example data record, of right-sided sciatic neuralgia is shown in Fig 4. A higher load in a heel zone, on an affected limb with lateralisation on a side of healthy limb was recorded. The burdened limb (in sciatic neuralgia) has relations to convexity of scoliosis.



Fig. 4. A case of a right-sided sciatic neuralgia

### Some Dynamic Measures and Characteristics

The limbs movement trajectories correspond with trunk abnormalities (Fig. 5). The limb load centre point moves behind the support zone.

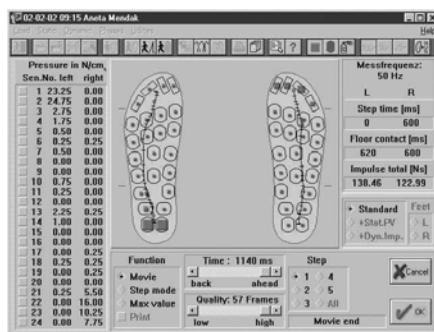


Fig. 5. The limb movement trajectories illustrations

In a step mode of dynamic data record visualisations same gait time intervals are visible. In Fig. 6 the example data map was presented, showing some medium values of a pressure flow.

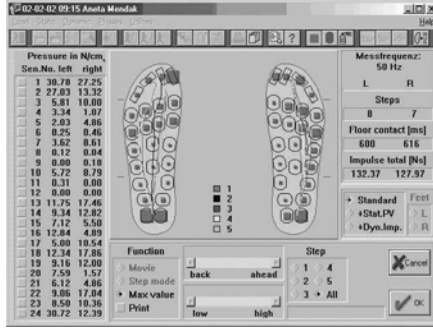


Fig. 6. The single step limb movement trajectories

The load centre movement trajectories displacement estimates the muscles tense, balance disturbances or paresis grade. The foot over load of the above case is visible on 3D map (Fig. 7).

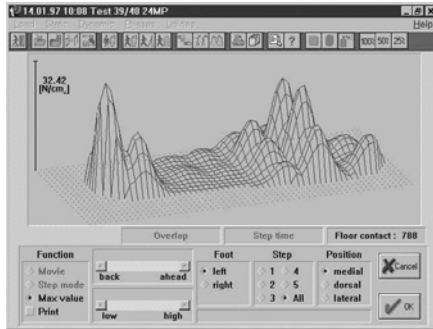


Fig. 7. The example 3D foot load distribution

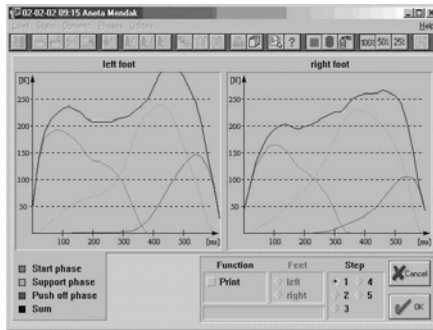


Fig. 8. A foot-load time diagrams

In Fig. 8 time diagrams of the gait load three phases are introduced: start phase, support phase and push-off phase. The over gone values, above physiology, have correlations with a pain regions. This option indicates the feet hooking on a floor, in push off or support phases, by non-burdened feet.

### The Body Weight Centre Translocation

The translocation of the body weight centre (gravity point) describes some instabilities observable in: trajectories shape and irregularities, their length and time relations. Irregular gravity centre displacement indicates a foot floor hooking cases noticed for strokes and a floor contact time elongation, characteristic for Parkinson’s disease.

In Fig. 9 the dynamic record illustrating the right-sided hemiparesis was presented. There are visible elongations of trajectories on paresis side (shorter on not affected side).

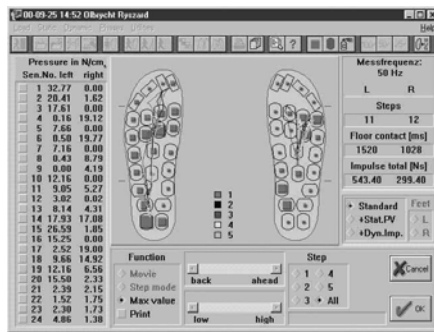


Fig. 9. Right-sided hemiparesis after cerebral stroke

Hemiparesis indications: the load centre located on not affected side; in majority cases in metatarsal region, with visible shorter trajectories. The affected limb is shifted from heel to toes irregularly that is visible as trajectories elongation, with disturbances at 1st and 2nd step phases.

Parkinson’s disease indications: bilateral fluctuations (tremor) of a load, usually dominated at one (an affected) side.

Sciatic neuralgia: with irregular and tangled fluctuations of a load (gravity) centre at the affected body side.

### Floor Contact Time and Impulse Relations

The floor contact illustrates values of impulse - bigger on neurologically affected side; respectively to the pathology level. Similar relations are noticed for impulse values of foot hitting a floor.

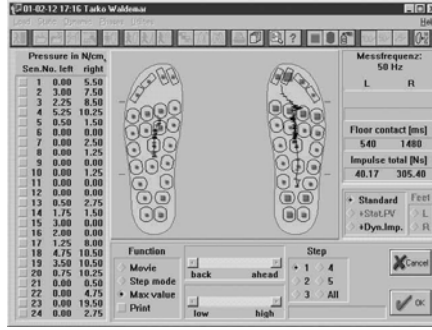


Fig. 10. The right-sided hemiparesis example impulses

In Fig. 10 the example record with right-sided hemiparesis was presented. The floor contact time is almost three times bigger on an affected limb with impulse over seven times bigger at the paresis side.

Hemiparesis: elongation of a step time on the affected limb, with significantly bigger impulse values, visible on the affected side, in a heel region and on not affected side, in metatarsus.

Parkinson’s disease: elongation of a floor contact time with impulse increase on neurologically affected side; the step time longer at the affected limb.

Sciatic neuralgia: elongation of floor contact time with impulse increase at the affected limb; the step duration time longer for a painful limb.

**Phases Time Relations**

The average times of start, support and push-off phases are used for step irregularities analysis. The start phase describes initial period, when the complementary limb is in push-off phase. The support phase shows the body weight going through the heel and metatarsus.

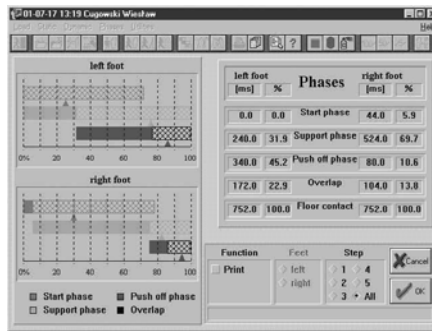


Fig. 11. The Parkinson’s disease phases example record



The right-sided lateralisation in Parkinson's disease has been presented in Fig. 11. There are noticed some extensions of the start and push-off phases, on the affected limb side.

Hemiparesis: with variable values of phases relations; the start phase elongations, shortening of the push-off and overlap phases on the affected limb.

Parkinson's disease: frequently with elongation of start and support phase on a side with neurological symptoms.

Sciatic neuralgia: irregular variability of phases' time relations.

## 4 Final Conclusion

The above gate descriptors identify the neurological diseases class, although more automatic recognition techniques are still under consideration. The trajectories movement illustrates many factors of diagnosis that can also be used for the disease monitoring, in rehabilitation processes.

The body weight goes into the paresis limb, shifting a load centre into affected limb. After medical treatment the decrease of muscular strength deficit was remarkable detected. The load (gravity) centre moved into a central line and into direction of the healthy limb. The trajectories are longer with diminished impulse on an affected foot.

On the affected side elongation of a start phase (with burdened heel) and a support phase (burden on metatarsus) were noticed. The dislocation of a body gravity centre and body balance disturbances were also clinically investigated and recognised.

After successful rehabilitation the above values were steadily coming back into normal ranges.

The sample analysis provided us with very essential conclusions of early Parkinson's disease diagnosis. Remarkably useful evaluation of specific pharmacotherapy processes were also carried out. The gait disturbances are estimated within their contribution proportions in final classifications.

The given gait analysis is necessary introduction into further investigations of automatic conclusions [4], [10] systems development.

## References

1. Zygula J., Piecha J., Gazdzik T. i inn. (1996) Zaawansowany system pomiarowy obciazen statycznych i dynamicznych dla diagnostyki schorzen ortopedycznych. *Chirurgia Narzadow Ruchu i Ortopedia Polska*, t. LXI 1996 suppl 3B, pp.118-124 (in polish)
2. Piecha J., Zygula J. (1995) PC Visual interface for Orthopaedic Expertise. Proc. of Int. Conference, pp. 162-167. Zakopane, May 1995.
3. Areal M., Brull M.A. (1976) A fundamental characteristic of the human body and foot, the foot ground pressure pattern. *J. Biomech* 1976, str.453-457.
4. Zygula J., Piecha J. (2006) *Journal of Medical Informatics and Technologies*, 2006.
5. Aritomi H., Morta M., Yonemoto K. (1983) A simple method of measuring the foot sole pressure of normal subject using Prescale pressure detectin sheets. *J. Biomech* 16, 1983, str.157-165.

6. Mumenthaler M. (1986) Diagnostyka rnicowa w neurologii. PZWL, Warsaw 1986 (in polish).
7. Limanowska K, Dega W. (1998) Rehabilitacja medyczna. PZWL Warsaw 1998 (in polish).
8. Chandzlik S., Piecha J. (2003) The body balance measures for neurological disease estimation and classification. *Journal of Medical Informatics & Technologies*, Vol. 6, pp: IT-87 – IT-94, 2003.
9. Zbrojkiewicz J., Piecha J. (2006) Gait characteristics features extraction for neurological diseases diagnostics. *Journal of Medical Informatics & Technologies*. ISSN 1642- 6037, 2006, Vol. 10, pp. 173-188.
10. Piecha J. (2001) The neural network selection for a medical diagnostic system using an artificial data set. *Journal of Computing and Information Technology CIT*, Vol.9, pp: 123–132, 2001.

---

# Analysis of Head-Mounted Wireless Camera Videos for Early Diagnosis of Autism

Basilio Noris<sup>1</sup>, Karim Benmachiche<sup>1</sup>, Julien Meynet<sup>2</sup>, Jean-Philippe Thiran<sup>2</sup>, and Aude G. Billard<sup>1</sup>

<sup>1</sup> Learning Algorithms and Systems Laboratory, EPFL, Switzerland  
basilio.noris@epfl.ch

<sup>2</sup> Signal Processing Institute, EPFL, Switzerland

**Summary.** In this paper we present a computer based approach to analysis of social interaction experiments for the diagnosis of autism spectrum disorders in young children of 6-18 months of age. We apply face detection on videos from a head-mounted wireless camera to measure the time a child spends looking at people. In-Plane rotation invariant Face Detection is used to detect faces from the diverse directions of the children's head. Skin color detection is used to render the system more robust to cluttered environments and to the poor quality of the video recording.

## 1 Introduction

Early diagnosis of Autism Spectrum Disorders is a central topic of research in developmental psychology. Several experimental protocols for the diagnosis and understanding of developmental disorders make use of video footage analysis to measure such elements as response time, attention changes and social interaction. However, up to now the analysis of such videos is processed by manually sifting through video frames and marking timestamps and events by hand. Several applications exist that help make this marking easier but the process is still heavily time consuming.

When trying to understand the social interaction abilities of the infant, one of the most informative cues is to observe whether she looks at the people surrounding her. This can be approximated by detecting faces in the head mounted video input. Face Detection has been a central research domain in computer vision for the last decade. Since the introduction of boosted cascades of weak classifiers[1], face detection has made a leap in terms of speed and performance and has become a robust tool for a large number of applications. Further works have improved this method by adding ulterior sets of features such as tilted Haar-like features[2], Gaussian filters[3] or Local Binary Patterns[4].

In this paper we recorded videos from a wireless camera placed on the head of children playing in an unconstrained environment to gather information on the amount of time they spent looking at the faces around them. We used an in-plane

rotation invariant face detection to take into account the wide range of possible orientations of the child's head. We combined it with skin color detection to limit the detection of faces to the zones of the image containing people.

## 2 WearCam

Recently we presented a head-mounted wireless camera that helps measure the visual attention of young children[5]. The WearCam (shown on Fig. 1) is a lightweight wireless camera mounted on the forehead of the child, thus giving a first-person point of view image that can give an approximation of the direction of attention of the child. The WearCam uses a TX45Light CCD sensor, designed to be used on miniature aircrafts, its lens has a diagonal field of view of  $92^\circ$ . The camera with wireless transmission electronics measures 27x27x38 mm. The sensor records an interlaced image of 640x480 pixels at 25 frames per second (fps). In interlaced videos, a single image contains two frames recorded at double framerate, effectively yielding a 50fps video, which is essential to sustain the child's head movement. However this reduces the resolution of the image by a vertical factor of two. The battery (8.4V NiMH rechargeable) is placed on the back of the head to balance the weight of the optics on the forehead. The weight of the whole camera, battery included, is 60 grams. The camera is designed to be worn by children aged between 6 months and 2 years. The head perimeter for children in that age range varies between 35 and 48cm<sup>1</sup>. Adjustable straps allow the WearCam to be fit on the head of the child. Optionally the WearCam can be attached to a cap to render it easier to place on the child's head.



**Fig. 1.** The WearCam, a 60 grams head-mounted wireless camera

## 3 In-Plane Rotation Invariant Face Detection

Most applications of face detection usually deal with upright frontal and out-of-plane rotated (profile) faces. Images are normally taken from stationary or level cameras and therefore the faces present in the images are rarely subject to strong in-plane rotations. With head-mounted cameras this is not necessarily

<sup>1</sup> Measures from the Swiss National Institute of Health.

true: the angle of view of a child can vary greatly depending on her position (sitting on the floor, on a tall chair, etc.) as well as the direction she is looking at (see Fig. 2).

Some technical challenges have to be taken into account when using the WearCam. *Wireless transmission* problems can appear as distortion and black-outs for a couple of frames or as a layer of noisy interference over the whole image, which can greatly decrease the performance of appearance/shape based detection methods. *The illumination* coming through the camera can undergo severe changes very rapidly as the child turns her head towards darker or brighter parts of the environment. This forces the video analysis to take into account a wide range of luminance variations. Additionally, *the sensitivity to color* can vary greatly depending on the amount of light present in the environment, making the chrominance response of the camera span from highly saturated colors to almost grayscale images.



**Fig. 2.** A normally developing child wearing a prototype of the WearCam looking at a mirror from a slanted angle

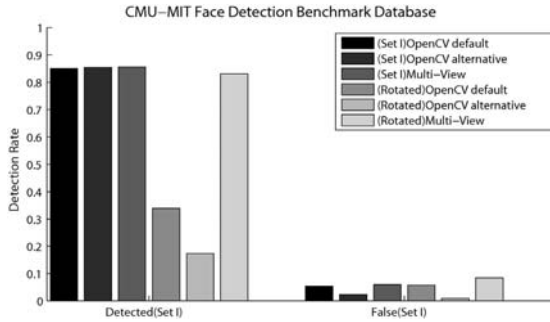
### 3.1 Boosted Cascades of Haar-Like features

The system presented here uses a set of 12 cascades of haar-like features trained on frontal faces at angles with steps of  $30^\circ$  of in plane rotations. Each cascade was trained from a set of 5000 rotated frontal faces and 3000 negative samples using the Gentle Adaboost algorithm, the sample size was set at  $20 \times 20$  pixels. Some random rotation was added to the face samples in order to span the  $30^\circ$  range covered by each cascade. The cascades were trained as part of [6]. The additional set of tilted features introduced by [2] was used for all angles except the  $0^\circ$ ,  $\pm 90^\circ$  and  $180^\circ$  angles where only the original set of Haar-like features was used. In each case the cascade is stump-based, with 21 to 23 stages, counting between 1480 and 1785 features depending on the angles. An additional cascade for profile faces is used to improve the range of detection. The profile cascade is part of the Intel OpenCV library<sup>2</sup>.

<sup>2</sup> Intel's Open Computer Vision Library, release v.1.0  
<http://sourceforge.net/projects/opencvlibrary/>

### Benchmarks

Tests on the CMU-MIT[7] benchmark were run for the multi-view face detector and compared with the default OpenCV cascades(Fig. 3). No improvement on the upright frontal faces (CMU-MIT Set I) is noticed. As expected, the detection of rotated faces (CMU-MIT Rotated) improves substantially. However, the amount of false positives increases.



**Fig. 3.** CMU-MIT face database benchmark. Comparison between the *default* OpenCV face detection cascade, the *alternative* OpenCV cascade (also part of the OpenCV distribution) and our combination of rotated cascades. A candidate region is considered valid if all features (eyes, mouth, nose) are inside the region and if the width of the region is less than twice the distance between the eyes.

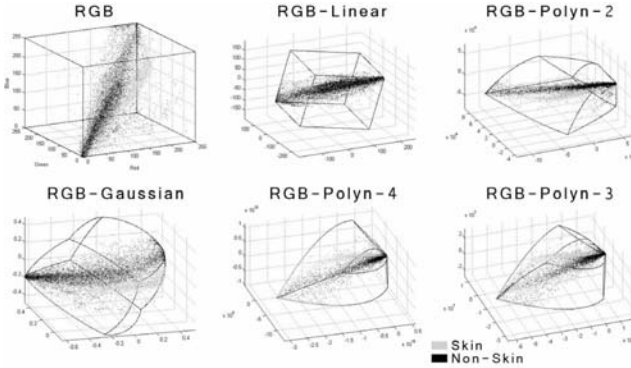
### 3.2 Pruning of False Detections

A drawback of using multi-view detection with a high number of cascades is that the amount of false detections increases. In cluttered environments such as children daycare centers this becomes an important limitation. In order to overcome this problem, skin color detection is used to mask the portions of the image which are more prone to false detection (e.g. colored drawings on the walls, bookshelves, etc.). The topic of skin detection has been extensively investigated in the past[8]. However no unanimous verdict as to which method and colorspace yields the best results exists and the methods should be chosen depending on the application.

The database we used for our tests was taken from [9]. The skin dataset contains 12'250'000 samples from people of different races and ages under varying lighting conditions, while the non-skin dataset contains 25'000'000 samples. Several methods were tested (Histograms, SVM, RVM and MLP) on different colorspace (RGB, HSV, YCbCr, SCT[10]). As most previous works[8] agree that dropping the luminance informations degrades the performance, we kept both the crominance and luminance information for all colorspace. Kernel-PCA was used to pre-process the skin and non-skin samples using linear, gaussian and polynomial kernels (Fig. 4 shows the projection of the RGB colorspace in different kernel spaces). The performance of the best classifiers and colorspace is given in Table 1.

**Table 1.** Classification error of the best performing methods and colorspaces for skin-detection

	SVM	RVM	MLP	HIST
RGB (poly3)	0.088	<b>0.059</b>	0.082	0.250
HSV (no-pca)	0.263	0.262	0.078	<b>0.058</b>
YCbCr (gauss)	<b>0.062</b>	0.080	0.092	0.180
YCbCr (no-pca)	0.137	0.130	0.074	<b>0.057</b>



**Fig. 4.** Example of Kernel-PCA projections of skin and non-skin samples

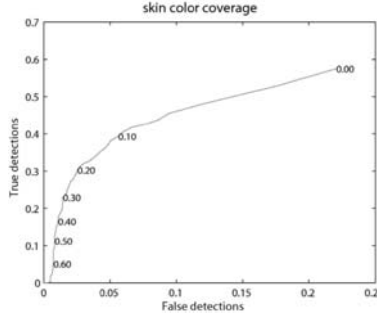
The performances of the different methods selecting their respective best colorspace are comparable. The best results were obtained using a histogram based classifier on the YCbCr colorspace. The solution presented in this paper uses a 64-bins histogram, the histogram resolution was obtained by cross-validation. Only the positive (skin) samples were used for training. For each video frame, a backprojection image of the skin histogram is computed using

$$BackProj(x, y) = hist_{skin}(y_{x,y}, cb_{x,y}, cr_{x,y})$$

where  $hist_{skin}$  is the skin histogram and  $y_{x,y}, cb_{x,y}, cr_{x,y}$  are the values of the pixel at position  $(x, y)$  in the image quantized to the histogram bin resolution. This yields a grayscale image that can be used as a mask for the face detection. If a face candidate does not contain a sufficient amount of skin pixels, the candidate is rejected. Fig. 5 shows the ROC curve of the skin pixels density necessary for a candidate to be considered as a valid detection. A threshold was set by elbow rule at 0.19. Due to the noisy nature of the input videos, no connectivity information on the skin color mask could be used successfully.

## 4 Experimental Setup

Recordings with the WearCam were made on 18 normally developing children (8 girls, 10 boys) between 2.5 and 4.5 years of age (mean 3 years and 4 months). The



**Fig. 5.** Receiver Operator Characteristic (ROC) curve for skin coverage percentage of face detection regions

children sat around a table in pairs, supervised by two adults, and were let play with the Sony Aibo<sup>3</sup> robot dog. The Aibo was set in its default pre-programmed behaviour, which involves responding to petting and stroking, obeying orders given through cards held in front of its head and playing with a colored ball or bone. Both children wore the WearCam and played for a duration of 8 to 14 minutes.

The ground truth data was generated by manually tracking faces in the videos using Adobe After Effects. The manual labelling took between 45 minutes and one hour for every minute of video footage, depending on the amount of head movement of the child and the number of faces appearing in the video. A total of 74511 faces were labelled in 62597 frames of video.

**4.1 Results**

Table 2 shows the results of running the system on 41 minutes of the recorded videos. The detection without skin pruning correctly found 55.4% of the faces, false detection amounted to 26.3% of all the detections. The detection using skin color pruning found 51.7% of the faces, reducing the false detections to 4.3% of all detections, thus reducing significantly the amount of false alarms while decreasing only slightly the performance of the detection.

**Table 2.** Results of face detection on 41 minutes frames of video

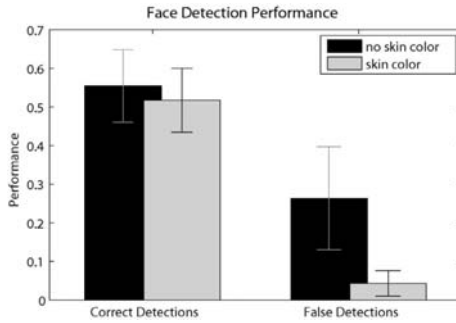
	faces detected	false detections
w/o skin pruning	55.4% ± 9.4%	26.3% ± 13.4%
skin pruning	51.7% ± 8.3%	4.3% ± 3.3%

**5 Discussion and Future Works**

We have presented a solution for the automatic analysis of videos from a head-mounted wireless camera for the evaluation of social interaction measured as the

<sup>3</sup> <http://www.sonydigital-link.com/aibo/>





**Fig. 6.** Face detection performance with and without skin color detection

proportion of time a child spends looking at faces. A rotation-invariant multi-view face detection using a boosted cascade of Haar-like classifiers was combined with histogram based skin detection to decrease false detections.

The system was able to detect more than 51% of the faces appearing in videos from free play in a cluttered environment. The skin color detection allowed to decrease the false detection rate by a factor of 6.

Compared to state of the art results the performance of our system might seem very low. However most systems run on data coming from constrained environments (e.g. uncluttered backgrounds, uniform illumination, stationary cameras) where the amount of movement or the quality of the image can be kept under check. Due to the very nature of the experimental goal (i.e. studying the behaviour of the infants in as unbiased a manner as possible) this is not possible in our case. The intense motion of the head, the wireless transmission noise and the sudden brightness and color intensity changes render the analysis of the WearCam videos challenging. The constraints the system must respect in terms of weight, dimensions and mobility do not allow the use of better quality cameras or wired transmission. Under these limitations, the results are at least promising.

One of the major issues of the system is that although mostly all the faces appearing in the video are detected fairly frequently, the detection is not continuous throughout the frames. Local spatio-temporal tracking of the detected faces should improve this and will be the main focus of future investigations.

**Acknowledgements.** This work was supported by the Thought in Action (TACT) project, part of the European Union NEST-Adventure Program, and by the Swiss Science Foundation within the National Center for Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2). The authors would like to thank Rainer Lienhart for his precious contribution and kind availability in providing trained cascades for face detection at different angles.

## References

1. Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
2. R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proceedings IEEE Conf. on Image Processing*, pages 900–903, 2002.
3. Julien Meynet, Vlad Popovici, and Jean-Philippe Thiran. Face detection with boosted gaussian features. *Pattern Recogn.*, 40(8):2283–2291, 2007.
4. Guillaume Heusch, Yann Rodriguez, and Sebastien Marcel. Local binary patterns as an image preprocessing for face authentication. In *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 9–14, Washington, DC, USA, 2006. IEEE Computer Society.
5. L. Piccardi, B. Noris, G. Schiavone, F. Keller, C. Von Hofsten, and A. G. Billard. Wearcam: A head mounted wireless camera for monitoring gaze attention and for the diagnosis of developmental disorders in young children. In *RO-MAN '07: Proceedings of the 16th International Symposium on Robot and Human Interactive Communication*, 2007.
6. R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. Mrl technical report, Intel Labs, Dec 2002.
7. Henry Rowley, Shumeet Baluja, and Takeo Kanade. Rotation invariant neural network-based face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1998.
8. P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122, 2007.
9. G. Gomez, M. Sanchez, and Luis Enrique Sucar. On selecting an appropriate colour space for skin detection. In *MICAI '02: Proceedings of the Second Mexican International Conference on Artificial Intelligence*, pages 69–78, London, UK, 2002. Springer-Verlag.
10. M.W. Powell and R. Murphy. *Position estimation of microrovers using a spherical coordinate transform color segmenter*. Fort Collins, CO, Jun 1999.

---

# Automatic Generation of 3D Lung Model

Dominik Spinczyk and Ewa Piętka

Institute of Electronics, Silesian University of Technology, 44-100 Gliwice,  
ul. Akademicka 16  
{dominik.spinczyk,ewa.pietka}@polsl.pl

**Summary.** The paper presents a method generating a 3D lungs model. The model is represented by a B-spline surface, generated by the global surface interpolation algorithm. The model to be derived is based on a vector of automatically obtained points marked on the contour of the lungs. Active contour method is used to make the lung segmentation more robust.

## 1 Introduction

The 3D models of anatomical structures supply the comprehensive description of anatomical features and enable communication between the image data and the symbolic knowledge. Very often models refer to the anatomy knowledge and High Resolution CT images. Zrimec et al. [1] have implemented Frame structures in a model consisting of a symbolic anatomical structure map and an annotated 3D anatomical atlas. In [2] a real-time physiologically-based breathing model of lungs under normal and pathological scenario has been conceived and implemented. Also a three-dimensional (3D) model of the human airway tree has been proposed based on a deterministic algorithm that can generate a branching duct system in an organ [3]. In the paper after the form of the curve the surface fitting method has been described (Section 2) the extraction of the vector of the points is discussed (Section 3).

## 2 Curves and Surfaces

### 2.1 Form of the Curve

The most common method of representing curves and surfaces in geometric modeling, which is also used in this study is a parametric form. In this form, each of the coordinates of the points on the curve is represented separately as an explicit function of an independent parameter:

$$\mathbf{C}(u) = (x(u), y(u)) \quad a \leq u \leq b. \quad (1)$$

There are several advantages of using parametric curves [5][6]:

- the parametric method can easily be extended to represent arbitrary curve in three-dimensional space,

- the parametric curves feature a natural direction of traversal – can be generated an ordered sequences of points along a parametric curve,
- the parametric form is more natural for designing and representing shape in a computer. The coefficients of many parametric form feature considerable geometric significance,
- the parametric form can represent a surface by introduction a second parameter.

**2.2 Types of the Curve**

The are several types of curves used in computer graphics. Curves consisting of just one polynomial segment are often inadequate. They require a high degree of order to satisfy a large number of constrains. For example n-order is needed to pass a polynomial Bezier curve through n data points. Also they are not well-suited to interactive shape design. As a solution can be used piecewise polynomial curves. This curve consists of  $n$  order polynomial segments:

$$C_i(u) = (x(u), y(u)) \quad 1 \leq i \leq m. \tag{2}$$

The curve  $C(u)$  is defined in  $u \in [0, 1]$ . The parameter values:  $0 = u_0 < u_1 < u_2 < u_3 = 1$  are called breakpoints. The segments are constructed so that they join with some level of continuity. The type of continuity does not need to be the same at every breakpoint. Any of the standard polynomial forms can be used to represent  $C_i(u)$ .

$$C_i(u) = \sum_{i=0}^n N_i(u)P_i \tag{3}$$

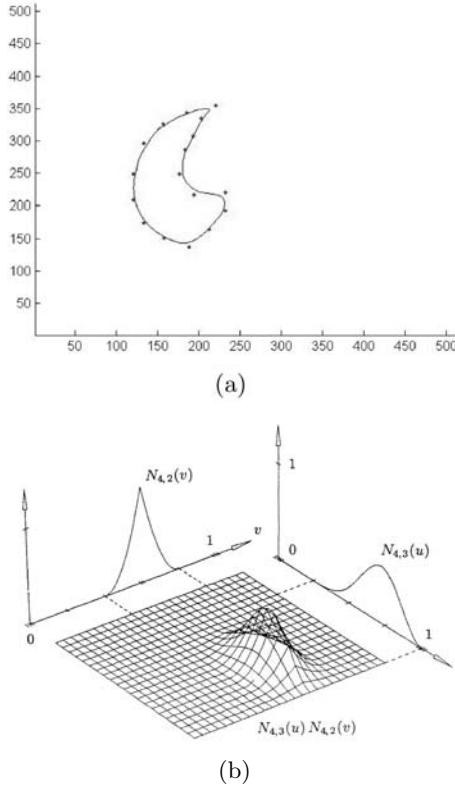
where:  $N_i$  - are basis B-spline functions and  $P_i$  are the control points. In every segment of the piecewise polynomial curve of  $n$  order only  $n$  basis B-spline function are not vanishing - this is called local support. The continuity is determined by the basis functions, hence the control points can be modified without alerting the curve's continuity [5]. A high degree of the curve is required to accurately fit some complex shape, yet high order curves are inefficient to be processed and are numerically unstable. Usually fourth order curves are employed(Fig. 1).

**2.3 Types of the Surface**

In the current study the tensor product surface is used. A B-spline surface is obtained by taking a bidirectional net of control points, two knots vectors, and the products of univariate B-spline functions [7] (Fig. 1):

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u)N_{j,q}(v)P_{i,j} \tag{4}$$

The cubic B-spline function  $N_{i,p} = N_{i,4}$  and  $N_{j,q} = N_{j,4}$  is used in each  $u$  and  $v$  direction.



**Fig. 1.** Example of (a) cubic curve for the left lung and control points (b) B-spline surface - cubic-quadratic basic function [5]

### 2.4 Surface Fitting Algorithm

There are two types of fitting surfaces [9]: approximation and interpolation. In interpolation, a surface which satisfies the given data precisely is constructed - the surfaces passes through the given points. In approximation, surfaces do not necessarily satisfy the given data precisely. In our study interpolation fitting and global interpolation algorithm are employed. Similar to the curve case, data points and degrees  $p$  and  $q$  are input. The interpolating a B-spline surface, two knot vectors  $\mathbf{U}$  and  $\mathbf{V}$ , one for each direction, and a set of points are required. Suppose the B-spline surface is given:

$$\mathbf{S}(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u)N_{j,q}(v)P_{i,j} \tag{5}$$

Since it contains all data points and since parameters  $s_c$  and  $t_d$  correspond to a data point  $D_{cd}$ , plugging  $u = s_c$  and  $v = t_d$  into the surface equation we obtain:

$$\mathbf{D}_{cd} = \mathbf{S}(s_c, t_d) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(s_c) N_{j,q}(t_d) P_{i,j} \tag{6}$$

$$\mathbf{D}_{cd} = \mathbf{S}(s_c, t_d) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(s_c) N_{j,q}(t_d) P_{i,j} = \tag{7}$$

$$= \sum_{i=0}^m N_{i,p}(s_c) \left( \sum_{j=0}^n N_{j,q}(t_d) P_{i,j} \right) = \sum_{i=0}^m N_{i,p}(s_c) \mathbf{Q}_{id} \tag{8}$$

### 3 Generation of the Model

The model to be derived is based on a vector of points marked in the image. This phase is performed automatically and employs a simple thresholding approach. The threshold value is found on the basis of a histogram analysis (Fig. 2). Two wave clusters can be seen in the histogram. The low value cluster reflects the background and lung areas, whereas the high value waves result from the tissue surrounding the lungs. The threshold value is found in the center of the longest plateau drawn on the level of one-fourth of the tallest wave in the lung area.

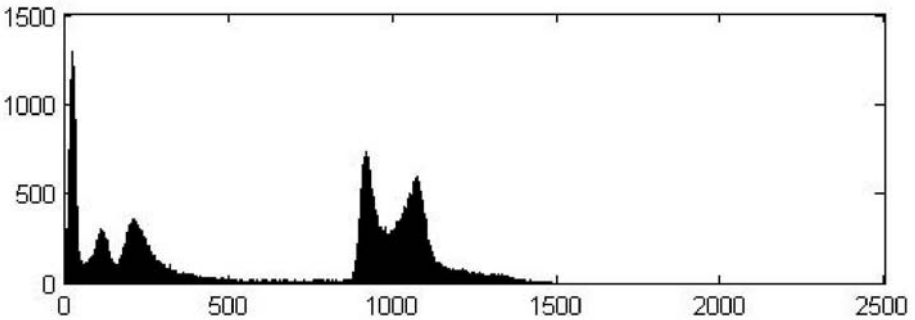
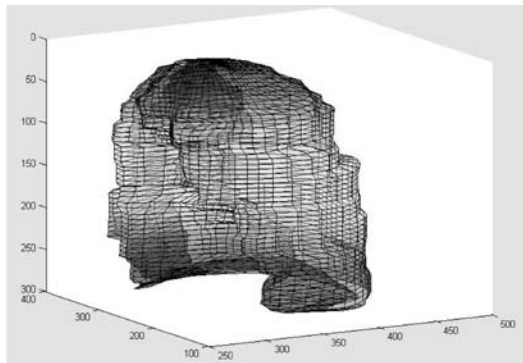


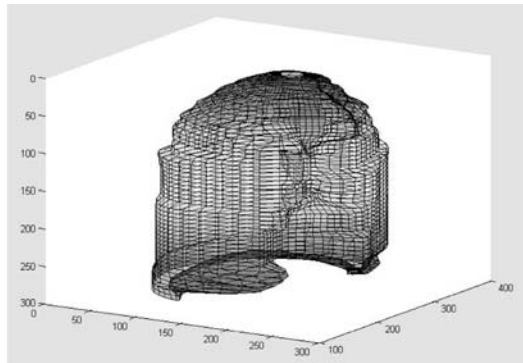
Fig. 2. Slice of CT chest and image histogram



(a)

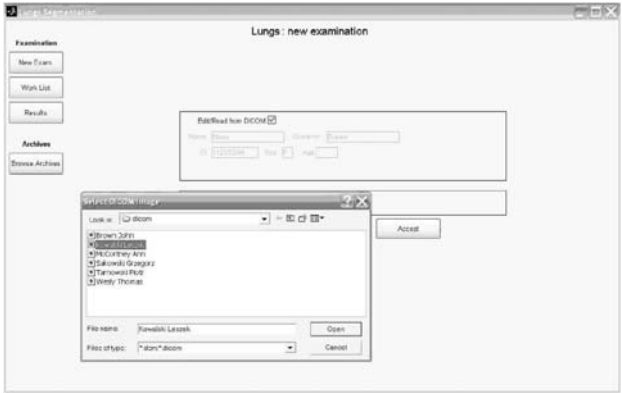


(b)

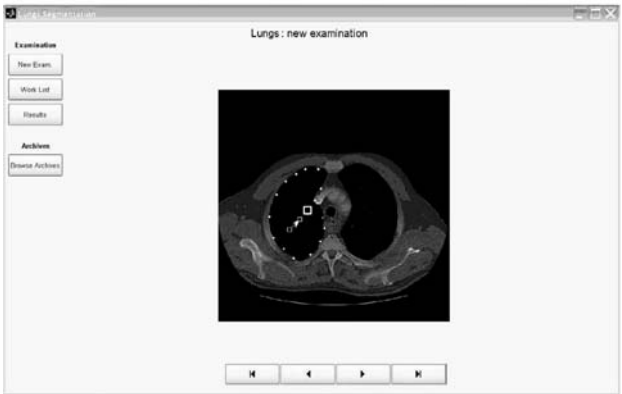


(c)

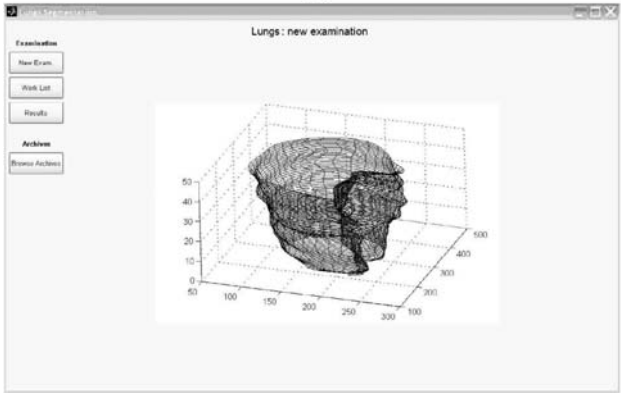
**Fig. 3.** (a) Vector of interpolation points in CT chest image (b),(c) the 3D model of left and rights lungs



(a)



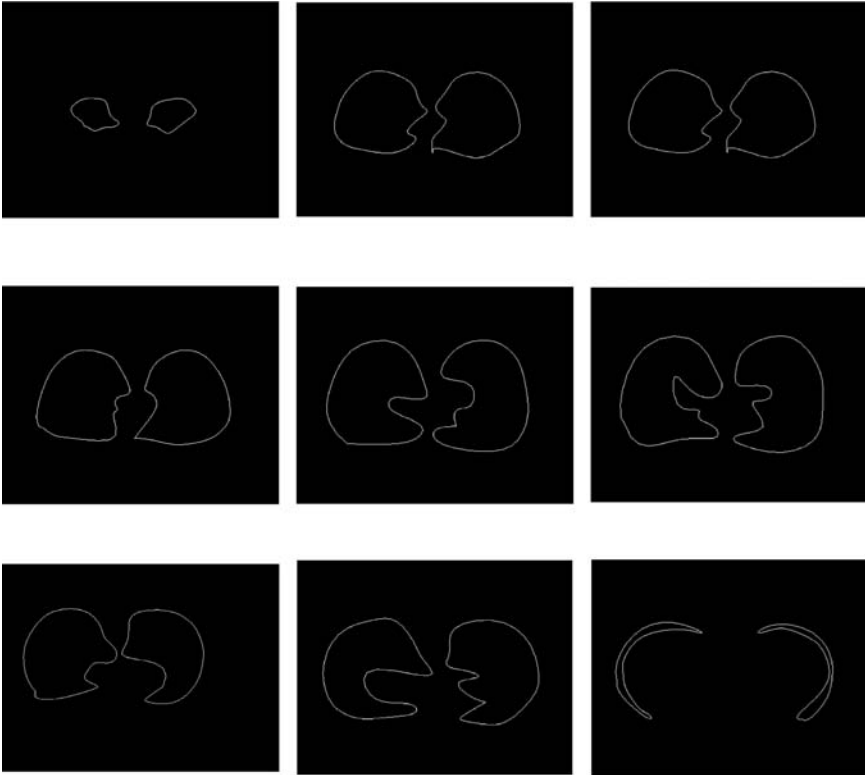
(b)



(c)

**Fig. 4.** (a) starting window in interactive editor (b) working area window (c) final lung model





**Fig. 5.** Curves of the left and right lungs

After the threshold is completed two largest objects, referred to as lungs, are selected. Then, the lung areas are extended to the contiguous slices. On each slice the border lines are extracted and points are equally distributed along them creating a vector of points (Fig. 3). This vector is subjected to the active contour analysis [10], which makes the lung segmentation more robust. The final vector of points has been used as an input to the surface fitting algorithm presented in Section 2. This results in a 3D model of the left and right lung (Fig. 3) and in 2D space (Fig. 4, 5).

## 4 Results

A graphical user interface has been developed in order to allow the user to test and verify the final result. The model, derived automatically on the basis of a patient data (Fig. 4a) is displayed in 2D space (Fig. 4b) for verification. If a correction is required, the point can be manually shifted and the entire set of new data subjected again to the surface fitting algorithm. The final version (automatically generated or manually corrected) is displayed in a 3D space (Fig. 4c). The

analysis has been performed on 10 CT chest exams of about 40 slices each. In 10 percent slices a manually correction has been required. The current study is part of the Computer Aided Diagnosis Workstation of volumetric measurement of pneumothorax project.

## References

1. Zrimec T, Busayarat S, Wilson P (2004) A 3D Model of the Human Lung, Medical Image Computing and Computer-Assisted Intervention - MICCAI 2004, LNCS, Volume 3217, pp. 1074-1075
2. Santhanam A, Pattanaik S, Rolland J, et al. (2003) Physiologically-based modeling and visualization of deformable lungs, Computer Graphics and Applications, 11th Pacific Conference, pp. 507 - 511
3. Suki B, Majumdar A, Takaki R, et al. (1999) A three-dimensional model of the human airway tree: application for airway-parenchymal interaction, Engineering in Medicine and Biology, 1999. 21st Annual Conf. and the 1999 Annual Fall Meeting of the Biomedical Engineering Soc. BMES/EMBS Conference, 1999. Proceedings of the First Joint, 1999
4. Bankman I (1999) "Medical Imaging", No. 9, pp. 519-530
5. Piegl L, Tiller W, Larobina M, et al. (1995) The NURBS Book, Springer-Verlag
6. Foley J, Dam A, Feiner S, et al. (1994) "Introduction to computer graphics", Addison-Wesley
7. Kiciak P (2005) Podstawy modelowania krzywych i powierzchni zastosowania w grafice komputerowej, Wydawnictwa Naukowo Techniczne
8. Surface Global Interpolation <http://www.cs.mtu.edu>
9. Dierckx P (1993) Curve and Surface Fitting with Splines, Clarendon Press
10. Spinczyk D (2005) Method of Active Contours in Segmentation of Anatomical Structures in MR Images", VII International Workshop for Candidates for a Doctor's Degree OWD'2005

---

# Semi-automatic Seed Points Selection in Fuzzy Connectedness Approach to Image Segmentation

Paweł Badura and Ewa Piętka

Silesian University of Technology, Institute of Electronics  
{pawel.badura, ewa.pietka}@polsl.pl

**Summary.** A new method improving the fuzzy connectedness approach to medical image segmentation is described. The segmentation based on fuzzy connectedness relies on a fuzzy connectivity scene creation by assigning a strength of connectedness to each possible path between some predefined seed point located inside an object and any other image element and performing the thresholding. The new idea is to automatically choose more seed points inside, as well as outside the segmented structure, in order to improve the method effectiveness and to reduce the computational time. The selection is based on two points marked manually.

The method has been tested on a set of 3D Computed Tomography (CT) lung images with delineated nodules. Examples and results of this method applied to the segmentation of lung nodules are shown.

## 1 Introduction

Segmentation of anatomical structures in medical images is one of the key points in a computer aided diagnosis (CAD). The quality of the segmentation strongly influences the reliability of CAD systems. The idea of conventional fuzzy connectedness [10, 8] has been used in previous studies [2, 4] in the 3D segmentation and visualization of cruciate ligaments of the knee joint in magnetic resonance (MR) images. In [3] the background seed point selection with adaptive thresholding has been used to limit the fuzzy connectivity scene computation process. The idea presented in this paper is to automatically choose a larger number of seed points inside and outside the object, based on two points marked manually.

This paper is organized in a following manner. First, the principles of fuzzy connectedness approach are introduced in section 2. Also, the relative fuzzy connectedness is presented in section 2.1 with a modification introduced by authors in earlier works [3]. In section 3 two algorithms of automatic seed points selection are described with example of their application. Short workflow of the entire algorithm is presented in section 4. Finally, section 5 includes the results obtained during evaluation of the method along with conclusions.

## 2 Fuzzy Connectedness

In the idea of fuzzy connectedness [10, 8] a fuzzy relation between two spels (spatial elements, pixels or voxels) within the image is defined. For any two elements  $c$  and  $d$  from an image  $C$ :

$$\rho = \{((c, d), \mu_\rho(c, d)) \mid c, d \in C \times C\}, \tag{1}$$

where  $\mu_\rho$  is a fuzzy membership function, so  $\mu_\rho \in [0, 1]$ . The relation is called [10] fuzzy spel affinity  $\kappa$ . For every two elements a value of  $\mu_\kappa(c, d)$  is assigned. It is based on their coordinate adjacency, intensities, gradient, and perhaps even their locations within an image.  $\kappa$  is reflexive:  $\mu_\kappa(c, c) = 1$  and symmetric:  $\mu_\kappa(c, d) = \mu_\kappa(d, c)$  for all  $c, d \in C$ .

The general form of  $\mu_\kappa(c, d)$  is:

$$\mu_\kappa(c, d) = \mu_\alpha(d, c) \cdot g(\mu_\phi(d, c), \mu_\psi(d, c)) \tag{2}$$

where  $\mu_\alpha$  is an adjacency relation (a hard 6-neighborhood in 3D or 4-neighborhood in 2D) and  $\mu_\phi, \mu_\psi$  represent the intensity-based, and intensity gradient-based parts of the affinity, respectively. Several possibilities for (2) have been described in [6]. A particular form of  $\mu_\kappa$  used in this study is:

$$\mu_\kappa(c, d) = 0.5\mu_\alpha \left( e^{\frac{(I(c)+I(d)-m_1)^2}{2\sigma_1^2}} + e^{\frac{(|I(c)-I(d)|-m_2)^2}{2(\sigma_2+m_2)^2}} \right) \tag{3}$$

where  $I(c), I(d)$  are intensities of points  $c, d$ , and  $m_1, \sigma_1, m_2, \sigma_2$  are parameters of  $\kappa$  related to intensities of the segmented object.

Fuzzy affinity as described above is nonzero only for the adjacent spels. We can call any pair of adjacent spels  $c, d$  a link, and the value of  $\mu_\kappa(c, d)$  – its strength. We call a path any sequence of spels  $\langle e_1, e_2, \dots, e_m \rangle$  such that for any  $i \in [1, m - 1]$  a pair  $\langle e_i, e_{i+1} \rangle$  is a link. It is noted  $p_{cd}$  if  $c = e_1$  and  $d = e_m$ . The strength of a path is the strength of its weakest link – the smallest affinity along the path:

$$\mu_N(p_{cd}) = \min_i \{ \mu_\kappa(e_i, e_{i+1}) \} \tag{4}$$

Finally, the fuzzy connectedness  $K$  is a fuzzy relation between spels  $c, d$  with a membership function  $\mu_K(c, d)$  being the strength of the strongest path  $p_{cd}$  of many paths forming a set  $P_{cd}$ :

$$\mu_K(c, d) = \max_{p_{cd} \in P_{cd}} [\mu_N(p_{cd})] \tag{5}$$

To perform the segmentation the fuzzy object  $\kappa\theta$  with affinity  $\kappa$  and a threshold  $\theta \in [0, 1]$  is defined as a set  $\mathcal{O}_{\kappa\theta}$  of spels such that for every  $c_1, c_2 \in \mathcal{O}_{\kappa\theta} : \mu_K(c_1, c_2) \geq \theta$ . First, a seed point  $o$  is chosen, which is supposed to belong to an object  $\mathcal{O}_{\kappa\theta(o)}$ . Then, the fuzzy connectivity scene  $C_o$  is computed for all  $c \in C$ :

$$C_o(c) = \mu_K(o, c) \tag{6}$$

with  $C_o(o) = \mu_K(o, o) = 1$  due to reflexivity of  $\kappa$ . Thresholding the fuzzy connectivity scene at some  $\theta$  gives a fuzzy object  $O_{\kappa\theta(o)}$ . A binary object  $O_{\kappa\theta(o)}$  – a result of the segmentation process – is defined as:

$$O_{\kappa\theta(o)}(c) = \begin{cases} 1 & \Leftrightarrow C_o(c) \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

For every two spels  $c_1, c_2 \in O_{\kappa\theta(o)} : \mu_K(c_1, c_2) \geq \theta$ , and for each spel  $d$  outside  $O_{\kappa\theta(o)}$  there exist some  $c \in O_{\kappa\theta(o)}$ , such that  $\mu_K(c, d) < \theta$ . Moreover, if  $c \in O_{\kappa\theta(o)}$ , then  $O_{\kappa\theta(c)} = O_{\kappa\theta(o)}$ . It means, that for fixed affinity  $\kappa$  and threshold  $\theta$  it does not matter which point from an object is taken as its seed.

It has also to be noticed, that, if a set  $O$  of  $M$  seed spels  $o_i$  is indicated, then the fuzzy connectivity scene  $C_O$  for such a set is defined as the union of the connectivity scenes for all  $o_i$ :

$$C_O(c) = \bigcup_{o_i \in O} C_{o_i}(c) = \max_{o_i \in O} [\mu_K(o_i, c)] \tag{8}$$

### 2.1 Relative Fuzzy Connectedness

The concept of relative fuzzy connectedness has been introduced in [5, 9]. A new object is indicated by marking its seed point  $b$  (or a set  $B$  of points  $b_i$ ) outside the  $O_{\kappa\theta(o)}$ . The new object  $O_{\kappa\theta(b)}$  is treated as a background. Two connectivity scenes are then determined for both, the object  $C_o$  and the background  $C_b$ . Finally, each spel is examined, whether its connectedness strength to  $o$  is greater than to  $b$ , i.e. if  $C_o(c) > C_b(c)$ , then  $c$  is included into  $O_{\kappa\theta(o)}$ :

This method does not require a threshold selection, yet it has also few disadvantages. It is sensitive to the seed points selection and fuzzy affinity definition. Moreover, it takes twice as much time as the conventional fuzzy connectedness approach. So, a modification of relative fuzzy connectedness method has been proposed [3]. Instead of creating the background fuzzy connectivity scene  $C_b$ , the object fuzzy connectivity scene  $C_o$  is computed until the point  $b$  (or  $b_i \in B$  with the highest  $C_o(b_i)$ ) is reached.  $C_o$  is nonzero only for spels  $d$  with fuzzy connectedness strength to  $o$  higher than between  $o$  and  $b$ , but not all of those spels belong to  $O_{\kappa\theta(o)}$ . One way to handle such “partial”  $C_o$  is thresholding [3]. In section 3.1 another approach is introduced: the introduction of a larger number of seed points  $b_i$ , located as close to the object boundaries as possible. They indicate the object and background features and lower the sensitivity to selected seed points (particularly if only single seed points  $o, b$  are chosen). Many seed points  $o_i$ , dispersed throughout the object, improve the determination of the fuzzy affinity  $\kappa$ . So, an algorithm of the object seed points  $o_i$  selection has also been implemented – as described in section 3.2.

## 3 Automatic Seed Points Selection

Since the overall goal of this study is to minimize the user interaction, only two seed points are requested to be pointed manually. One belongs to the object,

the other is located in the background. The sets  $O$  (object seed points) and  $B$  (background seed points) of seed points are created with manually indicated points  $o$  and  $b$  included in  $O$  and  $B$ , respectively.

### 3.1 Selection of Background Seed Points

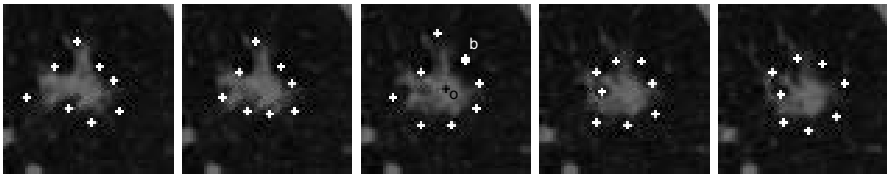
Three parameters are defined in the algorithm:

- intensity margin:  $m_I = |I(o) - I(b)|$ , where  $I(o), I(b)$  – intensities of  $o, b$ , respectively,
- distance  $r_{ob}$  between  $o$  and  $b$ ,
- angle  $\phi_{ob}$  between the horizontal ( $X$ ) direction and a vector  $\vec{ob}$ .

The steps of the algorithm are as follows:

1. A set  $\Phi$  of 8 angles uniformly distributed by  $45^\circ$ , starting at  $\phi_{ob}$ , is created.
2. Eight points  $b_i$  are searched on the seed slice in directions defined by angles  $\phi_j \in \Phi$ . The search starts at  $o$ . Indices of such points are generally real-valued, so they have to be rounded to the nearest integers. The pixel  $d$  at direction  $j$  (angle  $\phi_j$ ) is approved as a background seed point and included in set  $B$ , if its intensity  $I(d)$  is lower than  $I(b)$  (assuming that  $I(b) < I(o)$  – the object is brighter than the background) or in  $d$  there is an intensity minimum on line described by angle  $\phi_j$  and  $I(d) < [I(o) - 0.5m_I]$ . If a boundary of a slice is reached or a distance  $r_{od}$  exceeds  $3r_{ob}$ , than the direction  $j$  is left and no seed point is selected.
3. Eight points  $b_i$  per slice are searched on the neighbor slices, first up and then down from the seed slice (in  $Z$  direction). The search on each slice runs in the same way as on the seed slice, yet the central point is computed based on the previous slice – its central point and selected  $b_i$  points. The search is interrupted at slice  $k$ , if a central point for this slice satisfies conditions of being the background seed point, as shown at point 2. This point is included in set  $B$  as the only point from slice  $k$  and the analysis stops.

Fig. 1 shows an example of  $b_i$  points selection process at 5 CT lung slices with a nodule (only the region of interest – ROI – is shown).



**Fig. 1.** Selection of background seed points. A part of a 3D image with seed slice in the middle, seed point  $o$  marked with black  $+$ , points  $b_i$  with white  $+$ .

### 3.2 Selection of Object Seed Points

The evolutionary algorithm is used to find the set  $O$  of object seed points  $o_i$ . It operates in three dimensions with non-equal resolutions taken into consideration. There is no need to select points being the best representative of the object, according to the assumed criteria. It is sufficient to find points to be acknowledged as object elements. They should be well dispersed throughout the object, to create many centers for computing the fuzzy connectivity scene  $C_O$ . The evolutionary algorithm presented below satisfies these assumptions.

Here, the individuals are points (voxels), defined by their spatial coordinates. In every single run of the algorithm one point  $o_i$  is found and included in set  $O$ . The automatic selection of seed points is finished if the best individual in last generation of a single run of an algorithm has a fitness value (as defined in (9)) smaller than the minimal fitness  $fit_{min}$ , resulting from  $m_I$ . It means, that the number of seed points  $N_O$  is not fixed, yet is determined adaptively. Two parameters are computed:

- $r_{min} = \min \{ \max \{ 5, \overline{r_{B5\%}} \}, 15 \}$ ,
- $r_{max} = \max \{ r_{ob}, \overline{r_B}, 2r_{min} \}$ ,

where  $\overline{r_B}$  is the mean distance between the point  $o$  and all points  $b_i \in B$ ;  $\overline{r_{B5\%}}$  is the mean distance between point  $o$  and 5% of points  $b_i \in B$  being the closest to  $o$ . The features of an evolutionary algorithm are as follows:

- population size  $L_p$  equals  $r_{max}$ , and  $L_p \in \langle 20; 60 \rangle$ ;
- number of generations  $L_g$  is not fixed; the maximum of  $L_g$  is 50, but every 5 generations the fitness of the best individual is compared with the minimal fitness  $fit_{min}$ ; if it is greater, then the iteration terminates at current generation;
- the spatial coordinates of individuals creating the starting population are randomly selected from normal distributions with mean values equal to coordinates of point  $o$  and  $\sigma = 0.5r_{max}$ ;
- the ranked-based fitness assignment is used in the selection operation;
- in crossover operation the children are points lying between parents; mutation is performed by adding to the coordinate values some small random deviations (from  $N[0, 0.1r_{min}]$ );
- elitist strategy is used to save the best individuals in each generation; a small part of population is randomly generated in the same way as the starting population;

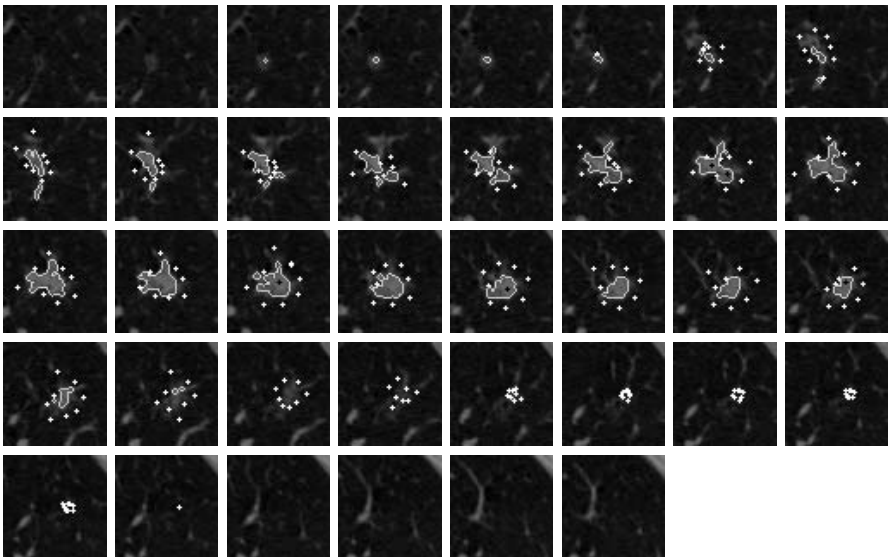
The fitness function for an individual  $p$  is described as:

$$fit(p) = \prod_{i=1}^4 f_i(p) \tag{9}$$

where  $f_i(p)$  are the components, defined as:

1. *Distance to  $o$  and  $o_i$ .* To avoid agglomerations of seed points, a basic rule is introduced: if the distance  $r_{o_i p}$  between a point  $p$  and  $o$  or any point  $o_i$  is smaller than  $r_{min}$ , then  $f_1(p)$  (and so  $fit(p)$ ) is set to zero. Otherwise  $f_1(p)$  decreases from 1 to 0 as  $r_{op}$  increases.
2. *Intensity  $I(p)$ .* The intensity component  $f_2(p)$  is represented by a normalized Gaussian distribution  $N[I(o), 0.3m_I]$ .
3. *Intensity profiles  $p - o_i$ .* The homogeneity of an object is evaluated by a component  $f_3(p)$ . First, a section  $P_{op}$  connecting  $o$  and  $p$  is defined with virtual points marked on it. For each virtual point (with real-valued coordinates) its intensity is computed using the trilinear interpolation. Subcomponent  $f_{3o}(p)$  decreases from 1 to 0 as two values increase: the intensity magnitude along  $P_{op}$  and the sum of absolute values of the difference between two adjacent points at  $P_{op}$ . Then, a similar subcomponent is computed for 3 points  $o_i$  of the closest distance to  $p$  (if they have already been determined). The final  $f_3(p)$  component equals to the maximum of subcomponents described above.
4. *Neighborhood of  $p$ .* To exclude points which are close to the object boundaries a neighborhood  $N(p)$  is taken into consideration. Intensity homogeneity and mean intensity of  $N(p)$  with reference to  $I(o)$ , influence  $f_4(p)$ .

An example presented in fig. 1 is expanded in fig. 2 to show all selected points  $o_i$  and  $b_i$ .



**Fig. 2.** Segmented object delineated on original image (ROI). A seed slice  $3^{rd}$  in the  $2^{nd}$  row, 5 selected points  $o_i$  marked with black +, 219 points  $b_i$  with white +.



### 4 Further Segmentation Process

Sets  $O, B$  are used to compute the fuzzy connectivity scene  $C_O$  as described in section 2. Then, a binary object  $O_{\kappa\theta(O)}$  is created according to (7), with threshold  $\theta$  being the smallest non-zero value in  $C_O$ . To avoid small holes within an object and smooth its edges a morphological 3D closing operation [7] is performed on  $O_{\kappa\theta(O)}$ , yielding a binary object  $O_{\kappa\theta(O)mc}$ . Edges of  $O_{\kappa\theta(O)mc}$  are then mapped on slices of the original image. The segmentation results are shown in fig. 2.

### 5 Results and Conclusions

A set of 17 CT lung studies have been used to evaluate the methodology. Images have been provided by the Lung Imaging Database Consortium (LIDC) [1]. They include lung nodules delineated by a group of radiologists and described in the form of a voxel-probability map  $M_{pr}$ : if all radiologists have indicated a voxel  $c$  to be a nodule, then  $M_{pr}(c) = 100\%$ ; if none has done so, then  $M_{pr}(c) = 0\%$  etc. Two methods have been compared – *RFC* (relative fuzzy connectedness with single seed points  $o, b$ , described in section 2.1) and *RFC-OB* (*RFC* with selection of seed points  $o_i, b_i$ ). The evaluation is based on the following measures:

- $P_{100\%}$  – percentage of correctly indicated voxels  $c$  with  $M_{pr}(c) = 100\%$ ,
- $P_{50\%}$  – percentage of voxels  $c$  assigned to a nodule by the algorithm with  $M_{pr}(c) \geq 50\%$ ,
- $P_0\%$  – percentage of false positives,
- $K_t = \frac{t_{RFC}}{t_{RFC-OB}}$  – computational time improvement factor.

**Table 1.** Comparison of the *RFC* and *RFC-OB* methods

	$P_{100\%} \pm \sigma_{P_{100\%}}$	$P_{50\%} \pm \sigma_{P_{50\%}}$	$P_0\% \pm \sigma_{P_0\%}$	$K_t$	$N_O$	$N_B$
RFC	97.7 ± 5.1%	74.4 ± 13.4%	7.6 ± 12.5%	3.4	1	1
RFC-OB	97.7 ± 5.1%	73.6 ± 12.5%	3.6 ± 5.4%	3.4	1	130.7
RFC	99.5 ± 0.8%	75.4 ± 13.4%	1.4 ± 2.6%	13.6	1	1
RFC-OB	99.8 ± 0.3%	77.8 ± 12.3%	2.0 ± 2.7%	13.6	13.8	296.1

Table 1 presents the results. For each image 5 pairs of seed points  $o, b$  have been indicated. For each pair the *RFC* algorithm ran once, the *RFC-OB* algorithm – 5 times. Table 1 includes also the mean sizes  $\overline{N_O}, \overline{N_B}$  of sets  $O, B$ . The images have been divided into 2 groups: a group including small nodules, where no additional  $o_i$  seed points have been selected (two top rows in table 1), and a group containing large nodules with  $\overline{N_O} > 1$  (two bottom rows). The  $\overline{K_t}$  factor has been computed for nodules with no leaks to other tissues in the *RFC* method.

The advantages of the *RFC-OB* algorithm appear especially in the case of a large structures. The improvement is noticeable in both: the  $M_{pr}$ -based factors (mainly because of the “*O*”-part of *RFC-OB*), and computational time (the “*B*”-part). The issue to be considered in the future is the affinity  $\kappa$  determination and its influence on the segmentation process.

## References

1. Ammato S et al (2004) Lung image database consortium: Developing a resource for the medical imaging research community. In: *Radiology* 232:739–748
2. Badura P (2006) 3D visualization of segmented cruciate ligaments. In: *XI International Conference: Medical Informatics and Technologies, MIT 2006* 69–74
3. Badura P (2006) Adaptive thresholding in fuzzy approach to segmentation of cruciate ligaments. In: *The Fifth International Workshop of Control and Information Technology, IWCIT 2006* 97–102
4. Badura P (2006) Metoda rozmytego podobieństwa w segmentacji więzadła krzyżowego tylnego. In: *VIII Międzynarodowe Warsztaty Doktoranckie, OWD 2006* 2:253–258
5. Herman G, Carvalho B (2001) Multiseeded segmentation using fuzzy connectedness. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23:460–474
6. Saha P, Udupa J, Odhner D (2000) Scale-Based Fuzzy Connected Image Segmentation: Theory, Algorithms, and Validation. In: *Computer Vision and Image Understanding* 77:9:145–174
7. Tadeusiewicz R, Korohoda P (1997) *Komputerowa analiza i przetwarzanie obrazów*. Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków
8. Udupa J, Saha P (2003) Fuzzy connectedness and image segmentation. In: *Proceedings of the IEEE* 91:1649–1669
9. Udupa J, Saha P, Lotufo R (2002) Relative fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24:1485–1500
10. Udupa J, Samarasekera S (1996) Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. In: *Graphical Models and Image Processing* 58:3:246–261

---

# Control of Dexterous Hand Via Recognition of EMG Signals Using Combination of Decision-Tree and Sequential Classifier

Andrzej Wolczowski<sup>1</sup> and Marek Kurzynski<sup>2</sup>

<sup>1</sup> Technical University of Wrocław, Faculty of Electronics, Institute of Robotics and Informatics  
andrzej.wolczowski@pwr.wroc.pl

<sup>2</sup> Technical University of Wrocław, Faculty of Electronics, Chair of Systems and Computer Networks  
marek.kurzynski@pwr.wroc.pl

**Summary.** The paper presents a concept of bioprosthesis control via recognition of user's intent. The set of elementary actions has been defined. We assume that each prosthesis operation consists of specific sequence of elementary actions. An example of prosthesis operations that can be composed into a decision tree is also presented. As a classifier the multistage recognition system is proposed, which combines sequential and decision-tree classifiers and its decisions are made on the basis of EMG signal analysis.

## 1 Introduction

The control of hand bio-prosthesis is usually based on myoelectric signals. Normally that kind of signals accompanies the muscles activity of healthy hand, but after the hand amputation they are (in greatest part) still available - along with the forearm muscles left in the stump. In the simplest case these signals can be detected (in a non-invasive way) on the surface of the skin using electrodes located above the examined muscles. That kind of measurement is called the surface electromyography (EMG) [1].

The EMG signal obtained in such a way is the sum of electrical phenomena taking place in the cells of the working muscles. Its form depends on the level of excitement and the spatial localization of the muscles, and that is how it identifies the type of the performed movement. This relation is the basis of the bio-prosthesis decision control. The patient tenses the muscles of the stump according to a prosthesis movement intention. The information about the type of muscles activity included in the EMG signals can be identified through adequate signal analysis (classification) [1], [8], [10], [13], [14], [15], [16].

To restore the lost efficiency of motion back at a satisfactory level, the bio-prosthesis should adapt the way of grasping to the type of grasped object. Such ability (called dexterity) is determined by two factors:

- the mechanical construction (the shape, dimensions, number of degrees of freedom),
- the possibility to control various actions (by action we understand the defined fragment of grasping/ manipulating motion - e.g. bending the finger in a joint).

A large repertoire of motion actions demands a large number of recognized classes of EMG signals. Furthermore, since the signal acquisition (especially performed in a non-invasive way, using electrodes located on the surface of the skin of the stump) is accompanied by numerous disturbances the reliable recognizing of the signals (especially when the number of possible classes is great) is a difficult problem. Reliability of EMG signal recognition (and thus the correctness of taking decisions by the system) is here the key issue as the prosthesis cannot perform any action inconsistent with human intent.

The paper presents a new approach to recognition of patient intent based on the treatment of motion actions as the elements of specified sequences and on the application of the decision tree classifiers, called multistage classifiers. The decision tree expresses the action logic of the classifier: the tree root is the start point for intention sequence recognition process, and each of permissible sequences creates the path from the root to terminal node representing the last intent in the sequence.

## 2 Model of the Hand

In an abstract approach, the human hand can be considered as a separate, goal-oriented system  $\langle H \rangle$ , while the nervous system controlling it can be considered as the control element  $\langle H_0 \rangle$  of this system [13]. The system operation consist in grasping the objects and, if need be, manipulating them (e.g. squeezing/pumping a rubber bulb, pressing the keys of cell-phone). A number of separate (elementary) actions can be distinguished in the process of the operation of such a system that repeat regardless of the type of a grasped object (type of grasp - see. fig. 2). These activities can be formally described as elements of a set:

$$\langle H \rangle \leftrightarrow A = \{a_1, a_2, \dots a_k\}. \quad (1)$$

Realization of each action  $a_i$  is a separate stage in the system operation, and can be interpreted as a discrete process state. In this case, the operation of system is a multi-stage discrete process, in which the change of state is related to the transition to the subsequent stage of the operation. "Observing" these states by classifying successive segments of EMG signal is the basis of multistage control of artificial hand [15].

In the simplest case, seven stages can be distinguished in the process of grasping with a hand:

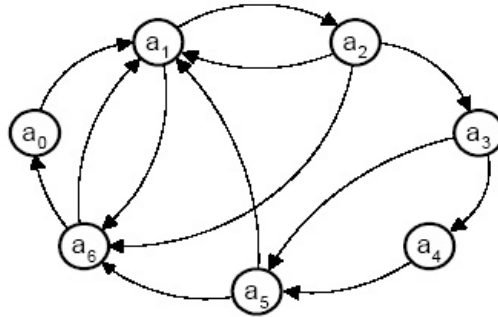
$a_0$  – rest position (starting point for the grasp preparation; the fingers stay at rest - half-closed arrangement, are motionless and passive),

- $a_1$  – grasp preparation (the fingers are "opening" or "closing", taking the posture depending on the shape of the object that is observed visually and the knowledge  $K$  about the method of grasping it, with the velocity proportional to the velocity of the intended arm movement),
- $a_2$  – grasp closing (precedes the grasp - the fingers move with the velocity resulting from the observation and knowledge about the object behaviour and the movement velocity during the previous stage, taking the posture depending on grasping object),
- $a_3$  – grabbing (the fingers squeeze the object with a force dependent on the knowledge and observed behaviour of the object and proportionally to the arm motion velocity in the grasp preparation phase),
- $a_4$  – maintaining the grasp (with force adjustment - the fingers increase/decrease the squeeze depending on object deformation and slip),
- $a_5$  – releasing the grasp (the fingers move with a velocity dependent on the knowledge of the object behaviour, e.g. small velocity for an object with an unstable balance),
- $a_6$  – transition to the rest position (the fingers move with a fixed velocity toward the rest position).

Between the states there exist some relations of admissible sequencing which enable the possible variants of the grasping process:

$$\Omega = \{(a_i, a_j) : a_i, a_j \in A\}. \quad (2)$$

These relations are shown in Fig. 1.



**Fig. 1.** A graph illustrating the admissible sequencing of the activities in the analysed grabbing process

### 3 Model of Control - Exemplary Decision Tree

Presented relations of action are valid regardless of the type of performed grasp (regardless of object being grabbed). The grasping process always begins and terminates with the rest position  $a_0$ . However, as follows from the description

above, the way of realisation of individual actions (excluding the rest position  $a_0$  and the transition to the rest position  $a_6$ ) connected with it, depends essentially on the type of performed grasp (i.e. on the object being grabbed). Next, different realization of the same actions (different way of moving the finger in the same grasping stage) implicate the different courses of EMG signals. Picking out these differences in connection with the knowledge of the actions sequencing relation constitutes the essence of multistage decision control (based on the decision tree). Multistage classification enables a better distinction of the intention of specific type grasp execution.

For the purpose of control, the described sequencing relation should be expressed in the form of the tree and should differentiate simultaneously individual actions depending on the type of grasp. Considering 10 types of grasp (according to Schlesinger classification) and the sequencing relation presented above, the tree dimension is large. To simplify the consideration we should take the following constraints - the number of the distinguished grasps should be reduced and the sequencing relation should be limited to the elementary sequence:  $a_0 - a_1 - a_2 - a_3 - a_4 - a_5 - a_6$ .

Let us consider the grasping of following objects: a pen and a credit card (standing in a container), a computer mouse and a cell phone (laying on the table), and a kettle and a tube (standing on the table).

## 4 Multistage Classifier of Elementary Actions

Let us consider seven-stage model of the activity of artificial hand presented in the section 2. Our task is to recognize at each stage elementary action, which a patient want to made. Let

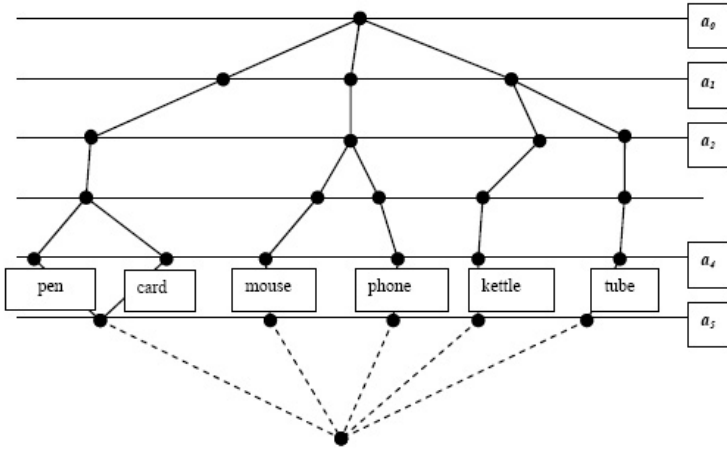
$$A^{(k)} = \{a_i^{(k)}\}_{i=1}^{m_k}, \quad k = 1, 2, \dots, 6, \quad m_6 = 1 \quad (3)$$

denotes set of classes (elementary actions) at  $k$ -th stage. Their number, character and practical meaning depends on the kind of bioprosthesis, its manual dexterity and possibilities which it offers. Some examples were given in section 3. Decision is made on the base of values of selected parameters of EMG signal (feature vector)  $x_{k-1} \in \mathcal{X} \subseteq \mathcal{R}^d$ , which are the same for all  $k$ .

Next, let us adopt probabilistic model of recognition task. It means, that features  $x$  are observed values of continuous random variable  $\mathbf{x}$  with conditional probability density functions (cpdf's)  $f(x/a_i^{(k)})$ ,  $k = 1, 2, \dots, 6$ ,  $i = 1, 2, \dots, m_k$ , whereas elementary actions  $a_i^{(k)}$  are realizations of random variable  $\mathbf{a}^{(k)}$ ,  $k = 1, 2, \dots, 6$ . We suppose next that between variables in the sequence  $(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{a}^{(3)}, \mathbf{a}^{(4)}, \mathbf{a}^{(5)}, \mathbf{a}^{(6)})$ , first-order dependence occurs, which is described by matrices of conditional probabilities:

$$P^{(k)} = [p_{a_i^{(k-1)}, a_j^{(k)}}]_{i=1,2,\dots,m_{k-1}, j=1,2,\dots,m_k}, \quad (4)$$

where



**Fig. 2.** Exemplary decision tree for patient intent sequence recognition  $a_0$  - rest stage;  $a_1$  - grasp preparation (3 different types of a finger gape);  $a_2$  - grasp closing (different for the kettle and the tube)  $a_3$  - grasping (different for the mouse and the phone - grasp and turn)  $a_4$  - maintaining the grasp (modification of grasp for the pen);  $a_5$  - releasing the grasp (equal for the pen and the card and different for the remaining items);  $a_6$  - transition to the rest position (different for all items)

$$p_{a_i^{(k-1)}, a_j^{(k)}} = P(\mathbf{a}^{(k)} = a_j^{(k)} | \mathbf{a}^{(k-1)}) = a_i^{(k-1)}. \tag{5}$$

Let us note, that for  $k = 1$  matrix  $P^{(1)}$  reduces to the vector of initial probabilities, namely:

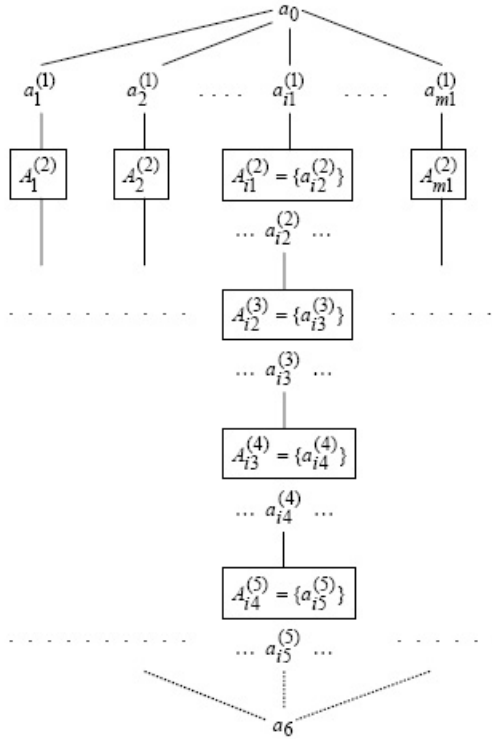
$$P^{(1)} = [p_{a_i^{(1)}}] = [P(\mathbf{a}^{(1)} = a_i^{(1)})]_{i=1,2,\dots,m_1}. \tag{6}$$

The whole multistage classification procedure can be conveniently described by means of decision-tree presented in Fig.3, in which interior nodes represent particular actions of classifier and their immediate descendant nodes denote sets of possible results of classification. These sets can be easily determined from (4), namely:

$$A_i^{(k)} = \{a_j^{(k+1)} : p_{a_i^{(k)}, a_j^{(k+1)}} > 0\}, k = 1, 2, \dots, 5. \tag{7}$$

Probabilistic model with first-order dependence between successive steps and decision-tree logic of multistage classifier, lead to the hybrid recognition system which combines characteristic features of the both approaches. In particular, it concerns decision rules (classification algorithms) for interior nodes of decision-tree. On the one hand classification algorithm at - let say -  $k$ -th stage ( $k = 1, 2, \dots, 6$ ) should take into account all available information contained in the sequence of observed feature vectors:

$$\bar{x}_{k-1} = (x_0, x_1, \dots, x_{k-1}), \tag{8}$$



**Fig. 3.** Decision-tree of seven-stage sequential classifier

what is typical for sequential classifier with dependences between patterns [5], [7]. On the other hand however decision at  $k$ -th step should consider the future into which this decision leads and furthermore it depends on the previous decision, what is characteristic for decision-tree classifier [4], [6], [12]. In the algorithm, such a ‘look ahead’ appears in the form of probabilities of correct classification at the next step of recognition. Combining both specific features of sequential and decision-tree classifiers, we get recognition algorithm at the  $k$ th step, which for probabilistic model is as follows:

$$\psi(\bar{x}_{k-1}, a_{i_{k-1}}^{(k-1)}) = a_{i_k}^{(k)} \text{ if} \tag{9}$$

$$p(a_{i_k}^{(k)} | \bar{x}_{k-1}, a_{i_{k-1}}^{(k-1)}) \times P(a_{i_k}^{(k)}) = \max_{j \in A_{i_{k-1}}^{(k-1)}} p(a_j^{(k)} | \bar{x}_{k-1}, a_{i_{k-1}}^{(k-1)}) \times P(a_j^{(k)}),$$

where  $P(a_j^{(k)})$  denotes probability of correct classification at the node  $a_j^{(k)}$ , whereas a *posteriori* probability  $p(a_j^{(k)} | \bar{x}_{k-1}, a_{i_{k-1}}^{(k-1)})$  can be calculated recursively from matrices (4) and cpdf’s [5].



## 5 Conclusions

Presented in this paper concept of hand bioprosthesis control is of preliminary nature. On the basis of set of elementary actions for exemplary six stage control procedure, a new recognition system has been proposed. Its idea consists in combination of sequential and decision-tree classifiers into common decision algorithm.

The basis for determination of classifying functions of recognition algorithm (9) is learning sequence containing a set of pairs: segment of EMG signal/the class of elementary action. Such a set can be experimentally determined by means of synchronous registering of movement of fingers and EMG signal. In order to collect learning set a measuring system is needed, which includes three sub-systems:

1. sub-system for measurement of EMG signal,
2. sub-system for measurement of hand movements,
3. sub-system which registers and integrates both kinds of data into representative learning set.

Concept of such a system can be found in [8].

**Acknowledgement.** This work was financed from the Polish Ministry of Science and Higher Education resources in 2007-2009 years as a research project No N518 019 32/1421.

## References

1. De Luca C. J. (1997) The Use of Surface Electromyography in Biomechanics, *Journal of Applied Biomechanics*, Vol. 13 (2) : 128-135
2. Duda R., Hart P., Stork D. (2000) *Pattern classification*, J. Wiley, New York
3. Eriksson E., Sebelius F., Balkenius C. (1996) *Neural control of a virtual prosthesis*, ICANN 96, Perspectives of Neural Computing, Springer Verlag
4. Kurzynski M. (1998) On the multistage Bayes classifier, *Pattern Recognition*, Vol 21 : 355-365
5. Kurzynski, M. (1998) Benchmark of Approaches to Sequential Diagnosis. In: Lisboa, P., Ifeachor, J., Szczepaniak P.S. (eds.): *Perspectives in Neural Computing*. Springer-Verlag, Berlin Heidelberg New York : 129-140
6. Kurzynski, M. (2004) Multistage Diagnosis of Myocardial Infarction Using a Fuzzy Relation. In: Rutkowski, L., Tadeusiewicz, R. (eds): *Artificial Intelligence in Soft Computing*, Lecture Notes in Artificial Intelligence, Vol. 3070, Springer-Verlag, Berlin Heidelberg New York :1014-1019
7. Kurzyński M., Zolnierek A. (2005) Computer-aided sequential diagnosis using fuzzy relations - comparative analysis of methods. ISBMDA 2005, *Lecture Notes in Computer Science*, Springer Verlag : 242-251
8. Kurzyński M., Wolczowski A., (2007) Recognition of patient intention via analysis of EMG signal, *Proc. 20th Conf. on Biocybernetics and Biomedical Engineering*, Wrocław 2007 (in press) (in Polish)
9. Laschi C., Dario P. et al. (2000) Grasping and Manipulation in Humanoid Robotics, *First IEEE-RAS Workshop on Humanoids - Humanoids 2000*, Boston

10. Nishikawa D. (2001) Studies on electromyogram to motion classifier, Graduate School of Engineering, Hokkaido University, Sapporo, Japa, 2001 (PhD Dissertation)
11. Quinlan J. (1990) Decision trees and decision making, IEEE Trans on SMC, Vol. 20 : 339-346
12. Safavian S., Landgrebe D. (1991) A survey of decision tree classifier methodology, IEEE Trans. on SMC, Vol. : 660-674
13. Wolczowski A. (2001) Smart Hand: The Concept of Sensor based Control, Proc. of 7th IEEE Int. Symposium on Methods and Models in Automation and Robotics, Międzyzdroje 2001
14. Wolczowski A., Kryzstoforski K. (2002) Control-measurement circuit of myoelectric prosthesis hand, Proc. of 13th Conf. of the European Society of Biomechanics, Acta of Bioengineering and Biomechanics 2002, Vol.4, Supp. 1 : 576-577.
15. Wolczowski A., Kurzynski M. (2004) Control of Artificial Hand via Recognition of EMG Signals, Lecture Notes in Computer Science 3337, Springer-Verlag Berlin Heidelberg : 356-367
16. Wolczowski A., Myśliński S. (2006) Identifying the relation between finger motion and EMG signals for bioprosthesis control, Proc. of 12th IEEE Int. Conf. on Methods and Models in Automation and Robotics, Międzyzdroje 2006.

---

# Posterior Cruciate Ligament – 3D Visualization

Piotr Zarychta

Silesian University of Technology, Institute of Electronics, Department of  
Biomedical Engineering, Akademicka 16, 44-100 Gliwice  
piotr.zarychta@polsl.pl

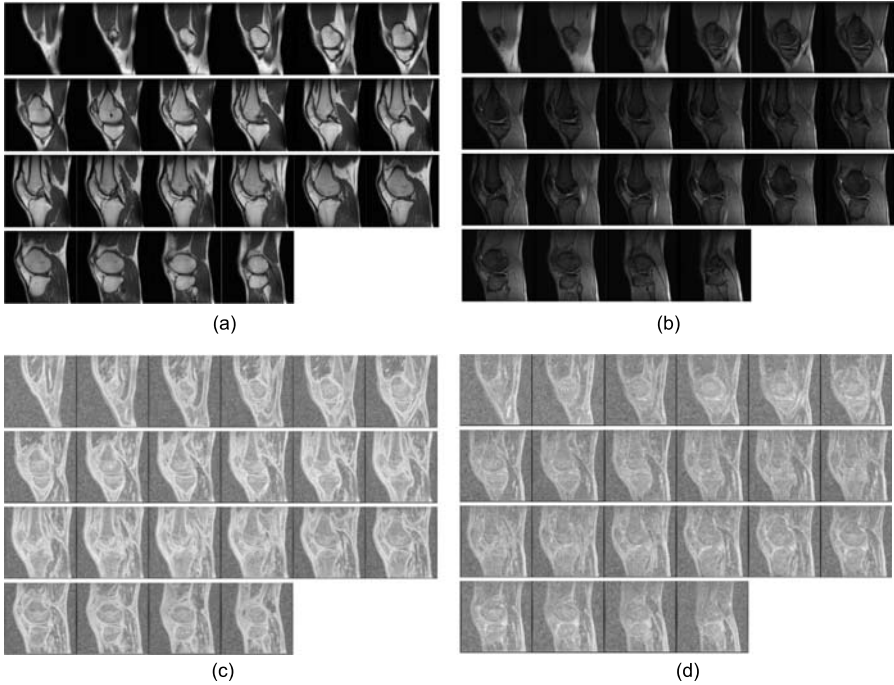
**Summary.** This paper shows a method which in the first step realizes an automatic registration of the T1- and T2-weighted MR knee images, in the second step locates the posterior cruciate ligament (PCL) on the T1-weighted MR knee images and in the third step permits 3D visualization of the PCL structures. The automatic registration process of the T1- and T2-weighted MR knee images is based on the entropy (or energy) measures of fuzziness. A method of location of PCL on T1-weighted MR knee images has been designed on the basis of both entropy (or energy) measure of fuzziness and fuzzy C-means (FCM) algorithm with median modification. The 3D visualization of the PCL structures procedure, which used registration and location steps, has been implemented in MatLab and converted to Visualization Toolkit (VTK).

## 1 Introduction

Posterior and anterior cruciate ligament (PCL/ACL) are anatomical structures which are (together with collateral ligament) responsible for the knee stability. Together with the shape of articular surface, muscles and contact forces they ensure proper arthrokinematics. During passive motion of the knee, cruciates help to change rolling into sliding movements and during active motion they resist translations and reduce shear forces. Cruciates control rotational movements in the flexed knee and together with collateral ligaments ensure rotational stability of the extended knee. For that reason cruciate ligaments belong to the group of anatomical structures, which are frequently susceptible to injuries, especially in the athletes. The success of a ligament reconstructive procedure depends on many factors, mainly accurate diagnosis based on the location and visualization of the cruciate ligament. The image analysis in knee ligament injuries is often performed on the T1- and T2-weighted MR images simultaneously. Slices of both series are read at the same scene. Thus, the image display should view the corresponding images of both series. In order to view matching slices next to each other, a registration may be required, if they are out of alignment.

## 2 Registration of the T1- and T2-Weighted MRI Images

Medical image registration is a difficult and complex problem. According to the database of the Institute of Scientific Information, in the last 15 years more



**Fig. 1.** MR knee images: original (a) T1-weighted and (b) T2-weighted, energy measure of fuzziness (c) T1-weighted and (d) T2-weighted

than 1000 papers have presented approaches to image registration [6]. In medical application, registration process should be automatic, reliable and easy to use [3]. T1- and T2-weighted MR knee images are recording at the same scene at different time. Sometimes patient motion during the medical examination can be a cause of a mutual shift of slices (fig. 1a,b). Although it is not a frequent case, the registration is required at the preprocessing stage.

The proposed registration process is based on a fuzzy image concept, which was introduced in [2] and implemented to biomedical signals in [1, 4]. That concept has been broadened to a two-dimensional case for the MR images [5], and then the final formula of entropy measure of fuzziness can be written as

$$H\left(\mu_{n,m,k}^\lambda(I_{i,j})\right) = F_1\left(\sum_{i=1}^{2k} \sum_{j=1}^{2k} h_\lambda\left(\mu_{n,m,k}^\lambda(I_{i,j})\right)p(I_{i,j})\Delta I_{i,j}\right), \quad (1)$$

where  $\mu$  is a membership function defined as in [5];  $p(\cdot)$  stands for a probability density function;  $h : [0, 1] \rightarrow R_+$  is an increasing function in  $[0, 0.5]$ , a decreasing function in  $[0.5, 1]$ , and  $h(0) = h(1) = 0$ ;  $F_1 : R_+ \rightarrow R_+$  is an increasing function and  $F_1(z) = 0$  if  $z = 0$ ;  $I_{i,j}$  denotes intensity of pixel  $I_{i,j}(n, m, k)$ ;  $\Delta I_{i,j}$  is a gradient between neighboring pixel values, and  $\lambda$  is a level of membership.

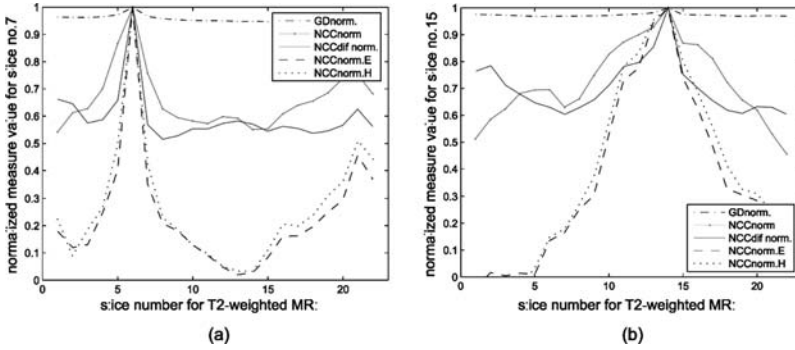


Fig. 2. Similarity measures of slice number (a) 7 and (b) 15

The introduction of the  $\lambda$  level into the measurement of fuzziness allows the insignificant membership degree to be removed.

The final formula for the energy measure can be expressed as

$$E\left(\mu_{n,m,k}^\lambda(I_{i,j})\right) = F_2\left(\sum_{i=1}^{2k} \sum_{j=1}^{2k} f_\lambda\left(\mu_{n,m,k}^\lambda(I_{i,j})\right) p(I_{i,j}) \Delta I_{i,j}\right), \quad (2)$$

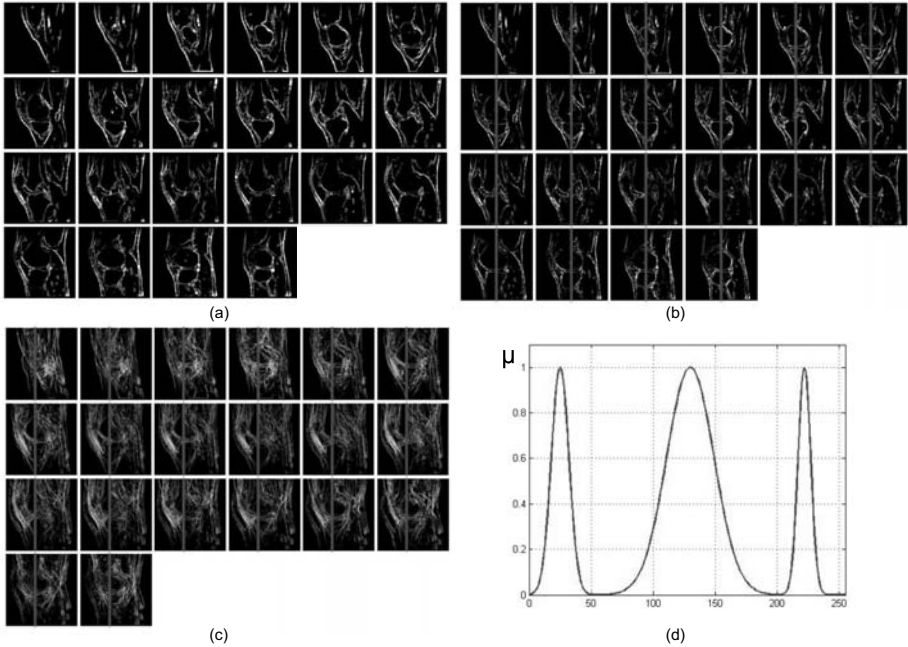
where  $f : [0, 1] \rightarrow R_+$  is an increasing function in  $[0, 1]$ , and for  $f(0) = 0$ ;  $F_2 : R_+ \rightarrow R_+$  is an increasing function and  $F_2(z) = 0$  if  $z = 0$ .

In order to evaluate the similarity of the current images, the following measures have been used: normalized cross correlation (NCC), gradient difference (GD) and gradient correlation (NCCdif) [5].

The registration process relies on comparison of each T1-weighted MRI slice with each T2-weighted MRI slice. The greatest similarity is expressed by the greatest measure value. The greatest measure value (fig. 2) denotes T2-weighted slice number, which is the most similar to given T1-weighted slice. In presented cases all measures give correct results, but measures for original images (GD, NCC, NCCdif) are sensitive to noise and interference. Two measures (for fuzzy images NCCnormH and NCCnormE) show strong discriminative power and are implemented in the application software.

### 3 Location of the PCL Structure on the MRI Images

The T1- and T2-weighted MR knee images, which were analysed, have size  $256 \times 256$  pixels. The average length of the side of a rectangle enclosing the PCL structure is equal about 100 pixels. In order to increase efficiency of computational procedures, it is very important to reduce the size to the region of interest (ROI 2D) including PCL structure. The proposed automatic method of location of the PCL on the T1-weighted MR knee images is based on entropy (or energy) measure of fuzziness and fuzzy C-means (FCM) algorithm with median modification. Standard FCM algorithm is widely used in many clustering



**Fig. 3.** T1-weighted MR knee images (a) FCM median modified, (b) FCM median modified with main axis, (c) centering and superposition operation with shown main axis (d) membership function

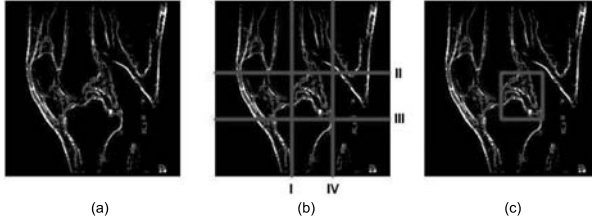
approaches. Its advantages include a conceptual and computational simplicity and the ability to model uncertainty within the data. FCM has also several weaknesses. It does not incorporate spatial context information which makes it sensitive to noise and image artifacts.

In this paper a modified method of FCM is used for the clustering. The FCM objective function is modified by adding a second term, which formulates a spatial constraint based on the median estimator. Then, the final formula for FCM median modified can be expressed as

$$J(\mathbf{U}, \mathbf{V}) = \sum_{k=1}^K \sum_{n=1}^N u_{kn}^{r_f} (\|\mathbf{x}_n - \mathbf{v}_k\|^2 + \alpha \|MedF(\mathbf{x}_n, Z) - \mathbf{v}_k\|^2). \quad (3)$$

where  $MedF(\mathbf{x}_n, Z) = median(S)$ ,  $S = neighbourhood(\mathbf{x}_n, Z)$  and  $Z$  determines the size of the mask. The processed clinical MR knee T1-weighted after fuzzification with the FCM median modified is shown in the fig. 3a.

On the basis of the analysis of each profile in every slice of the MR knee T1-weighted after fuzzification and fuzzy clustering, a main axis running along the thighbone and tibia is determined (fig. 3b). The main axis is determined according to following formula



**Fig. 4.** Location of the PCL in the knee: (a) slice no.12 from fig. 3a, (b) marked four axis, (c) marked ROI 2D

$$th = \min_k [L_{pr}(k)], \text{ for each } L_{pr}(k) \neq 0, \tag{4}$$

where  $L_{pr}(k)$  denotes number of nonzero values of the pixel intensity in the  $k$  profile. When several profiles meet condition (4), the profile of the lowest nonzero pixel value is chosen. In the next step for teaching group (56 cases) the centering and superposition operation is implemented (fig. 3c). Profile analysis after the centering and superposition operation permits to create the membership function (fig. 3d). This stage gives the information, which is indispensable for the formation of the membership function (location and size of the stripes in the main axis). On the basis of thus analysis a membership function has been determined, according to following formula

$$\mu(i) = \sum_{n=1}^3 \exp\left(\frac{-(i - k_{srn})^2}{2\sigma_{pn}^2}\right), \tag{5}$$

where  $\sigma$  is a standard deviation and  $k$  denotes an average column number in the specific range. Then, edges of a ROI 2D containing the PCL are found according to following rules:

- **Left edge** (axis I in the fig. 4b)

$$\min_k \left[ \min_j [\bar{\mu}(i), pr_{i(k)}^j] \right], \tag{6}$$

where  $k$  - profile number in the slice,  $j$  - slice number in the image,  $\bar{\mu}$  - complement of the membership function,  $pr_{i(k)}$  - is a  $i$ -value of a  $k$ -profile,

- **Upper edge** (axis II in the fig. 4b)

$$\max_u [d_u] \tag{7}$$

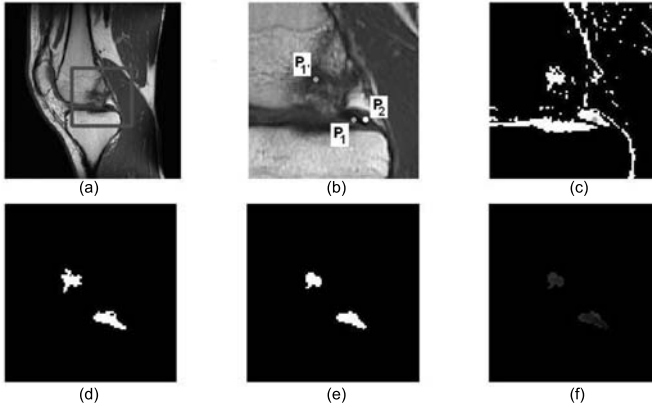
where  $d_u$  - distance between I axis and right edge of thighbone for  $u$ -profile,

- **Lower edge** (axis III in the fig. 4b)

$$\max_p [d_p] \tag{8}$$

where  $d_p$  - distance between I axis and right edge of tibia for  $p$ -profile,

- **Right edge** (axis IV in the fig. 4b) has been determined on the basis of the axis I shifted by (8).

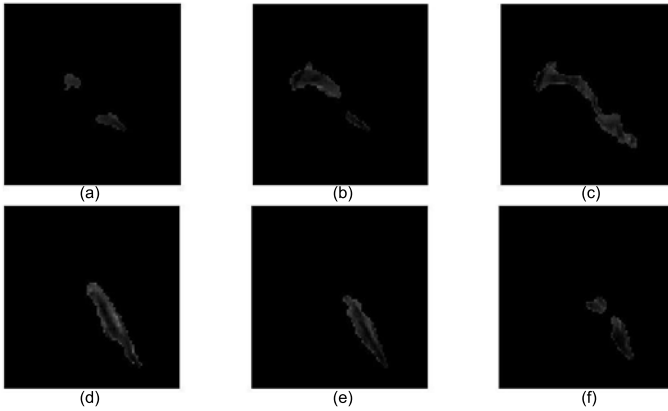


**Fig. 5.** Stages in the PCL structure build

#### 4 3D Visualization of the PCL Structure

The finding of the ROI 3D including PCL of the knee joint is based on the mapping ROI 2D on the all MR knee T1-weighted slices. Curve analysis depiction of the entropy (or energy) measures of fuzziness in ROI 3D gives information on the top limit, and histograms analysis in the ROI 3D about the bottom limit. Both the top and bottom limits determine slices including PCL structures. Depending on slice thickness PCL structure can be present on 4-6 slices. Identification of these slices is permitted at the beginning of computer processing. The goal of that processing is to get the PCL structures from the MR knee T1-weighted slices (fig. 5). The first step of this process is based on the plotting of the determined slices including PCL structures ROI 2D (fig. 5a). In the second step the radiologist should insert two points: the first one (point  $P_1$  in the fig. 5b), which belongs to PCL and the second one (point  $P_2$  in the fig. 5b), which does not belong to PCL. After the inserting those two points, the fuzzy clustering with adaptive matched number of different clusters is performed. The next step of this process depends on the analysed ligament structure. If two separate ligament structures are situated on the analysed slice (thigh attachment and tibia attachment), then radiologist has to insert an additional point (point  $P_{1'}$  in the fig. 5b). The next stage of the 3D construction process is based on choosing from the clustering image (fig. 5c) objects which include PCL structures and eliminating the other. To this end, the coordinates of the points inserted by the radiologist ( $P_1$  and  $P_{1'}$ ) were carried on the clustering image. Then only objects including these coordinates were left (fig. 5d). In order to smooth edges and fill possible holes in an object (or objects) which was received in a prior stage of this process, were used close and open operation (fig. 5e). The last step of that process is based on the plotting of the smoothed and filled object on the input image (fig. 5b). The logical product of these images is shown in the fig. 5f. This image is one of the base images received for one patient. The rest of these base





**Fig. 6.** Base images received for one patient



**Fig. 7.** Selected projection of the 3D PCL structure

images have been received after using the analogous operation described above. The set of the base images for one patient is shown in fig. 6 and on the base of these images a 3D PCL structure has been built. This spatial structure of the PCL (fig. 7) was created through layer composition of the base images by using a linear interpolation. In order to create a 3D visualization of the PCL structure the Visualization Toolkit (VTK) was used. VTK was chosen first of all because this software is independent of the platform (Windows, Unix) and makes it possible to start the program on any computer (for example portable computer). This is very important for the radiologist because he can show the concrete clinical case not only in the hospital computer laboratory.

## 5 Numerical Results and Conclusion

The methodology has been tested on 116 clinical T1- and T2-weighted MR knee studies. In all cases the registration process has been performed correctly. In 110 (94%) cases the proposed method location of the PCL on the T1-weighted MR knee images has been executed correctly. The visualization process was carried out correctly in all cases, where the location of the PCL structures on the T1-weighted MR knee images was marked correctly, but unfortunately the

process based on the T2-weighted MR knee images was carried out incorrectly in 8 from 110 cases (although few operations of the contrast improvement were used). Performance impossibility of the 3D PCL structure on the basis of the T2-weighted MR knee images was caused through lack of uniform, intense signal in the ligament structure. Mistakes connected with localization process were caused by following factors: a pathology depends on the ligament breaking together with a fragment of the thighbone (2 cases), a lack of the location of thighbone and tibia along the same axis (3 cases) and a location of the knee joint in the extreme part of the slice (1 case).

Software was built on the basis of described methodology, which is dedicated for the PCL diagnosis and makes it possible to show PCL structure in 3D space. The proposed method of broadening the idea of a fuzzy image creating and using the entropy and energy measure of fuzziness to the registration of the T1- and T2- weighted MR knee images, and to the location and finally to the 3D visualization of the PCL, seems to be very effective and promising.

## References

1. Czogała E, Łęski J (1997) Application of entropy and energy measures of fuzziness to processing of ECG signal, *Fuzzy sets and systems*:9–18
2. Deluca A, Termini S (1972) A definition of non-probabilistic entropy in the setting of fuzzy set theory, *Information and Control*, 20:301–312
3. Nalecz M (2003) *Obrazowanie Biomedyczne. Biocybernetyka i Inżynieria Biomedyczna*. Exit, Warszawa
4. Rozentryt P, Czogała E, Łęski J (1996) Application of entropy and energy measures of fuzziness to heart rate variability analysis, *Medical Science Monitor* 2(5):642–649
5. Zarychta P (2006) Location and 3D visualization of the cruciate ligament in the MR knee images on the basis of fuzzy logic. PhD Thesis, Silesian University of Technology, Gliwice
6. Zitova B, Flusser J (2003) Image registration methods: a survey, *Image and Vision Computing* 21:977–1000

**Various Applications**

---

# A Self-training Approach for Automatically Labeling IP Traffic Traces

Francesco Gargiulo, Claudio Mazzariello, and Carlo Sansone

Università di Napoli Federico II, Dipartimento di Informatica e Sistemistica,  
via Claudio 21 — 80125 Napoli, Italy  
{francesco.grg, cmazzari, carlosan}@unina.it

**Summary.** Many approaches have been proposed so far to tackle computer network security. Among them, several systems exploit Pattern Recognition techniques, by regarding malicious behavior detection as a classification problem.

One of the main problems occurring in real environments is the difficulty in collecting a representative dataset of traffic traces which is correctly labeled, so that it can be used for reliably training supervised techniques. To overcome this problem, we present a self-training architecture based on the Dempster-Shafer theory for automatically building a database of labeled traffic traces from raw tcpdump packets. We also show that intrusion detection systems trained on such a database perform approximatively as well as the same systems trained on correctly labeled data.

## 1 Introduction

As attack strategies become more and more sophisticated, the task of detecting attacks or malicious activities in network traffic becomes harder and harder. Techniques explicitly modeling malicious behavior realize the principle at the basis of *misuse detection* [7] approach. Within this category fall also Intrusion Detection Systems (IDS) that search for well-known attack signatures in network traffic, i.e. *signature-based* IDS. The completeness request is actually their major limit. On the opposite, *anomaly-based* techniques model the expected normal behavior of the system. Such technique can detect unknown intrusions (the so-called *zero-day attacks*); the difficulty is that they should be general enough to allow authorized users to work without raising false alarms, but specific enough to recognize unauthorized usages.

In [7] it is shown how the network intrusion detection problem can be easily formulated as a typical classification problem: given information about network connections between pairs of hosts, the task is to assign each connection to one out of two classes, respectively representing normal traffic conditions or the attack category. The main advantage of a Pattern Recognition (PR) approach is the ability to generalize which is peculiar to PR systems. They are able to detect some novel attacks, without the need of a complete description of all the possible attack signatures, thus overtaking one of the main drawbacks of

misuse detection approaches, especially those implemented by means of signature detection techniques. Typically, PR approaches imply the use of IDS based on a misuse detection approach [7], but also anomaly-based systems implementing unsupervised techniques have been considered [14]. Anyway, it is worth noting that an IDS realized by means of a PR approach based on supervised learning techniques typically performs better than those based on unsupervised learning techniques [11].

One of the main drawbacks occurring when using PR-based system in real environments is the difficulty in collecting a representative labeled set of data for training them. This aspect is often disregarded in papers proposing PR approaches for coping with the intrusion detection problem, where usually the well-known *KDD 99* [3] database is used for testing [15]. This choice is no longer accepted in the IDS community, since network traffic is now significantly changed and some papers (see for example [12]) highlighted many flaws in the construction of such a database.

Hence, in this paper we propose a general methodology for automatically obtaining a dataset of labeled packets starting from raw tcpdump traces. Such dataset can be then used for the training phase of supervised PR algorithms. As we propose to use multiple detection techniques, it is necessary to employ some information fusion strategy in order to combine them. Since we have no prior knowledge about traffic characteristics, we use the Dempster-Shafer [8] theory. We will also evaluate to which extent a dataset labeled by using our approach can effectively be used to train a PR-based IDS. Moreover, we will show that it is also possible to integrate a trained PR-based IDS in the proposed architecture, in order to obtain a database labeled in a more reliable way. It is worth noting that our work can be also regarded from the perspective of semi-supervised classification. Some analogies can be found between the *soft* self-training techniques described in [16] and the architecture for automatic labeling described in the rest of the paper.

The paper is organized as follows: in section 2 we will present the architecture aimed at allowing the automated labeling of raw packets from tcpdump traces. Then, in section 3 we will illustrate how an IDS trained on such a database can be integrated into this architecture in order to give rise to a more reliably labeled dataset. In section 4 we will describe our tests; we first show that with the proposed architecture it is possible to obtain a reliably labeled traffic dataset. Then, we will show that systems trained on such dataset perform approximatively as well as the same systems trained on correctly labeled data. Finally, in section 5 some conclusions are drawn.

## 2 A Self-training Architecture for Building a Training Dataset

In network security, there are not too many issues in collecting large amounts of traffic which might be used to train a supervised classifier. Unfortunately, such traffic often lacks a reliable labeling. In order to overcome this problem,

we propose a self-training architecture for automatically building up a traffic database containing packets each labeled either as *normal* or as an *attack*. The proposed architecture consists of several base classifiers, called Base IDS (B-IDS in the following). According to the definition of our framework, B-IDS must not require to be trained on labeled data. Typical examples are signature-based IDS (such as Snort<sup>TM1</sup>) or, more in general, any IDS based on unsupervised techniques. As we propose to use multiple detection techniques, it is necessary to employ a fusion strategy in order to combine their outputs.

## 2.1 The Dempster-Shafer Theory for Combining Multiple B-IDS

The theory of Dempster and Shafer (D-S theory) has been frequently applied to deal with uncertainty management and incomplete reasoning. Differently from the classical Bayesian theory, D-S theory can explicitly model the absence of information, while in case of absence of information a Bayesian approach attributes the same probability to all the possible events. Moreover, D-S theory disregard any prior knowledge about the data to be classified. The above described characteristics can be very attractive for managing the uncertainties of the network security domain, due, for example, to the presence of zero-day attacks.

According to the D-S theory, a *basic probability assignment* (*bpa*) has to be associated to each base classifier, which describes the subjective degree of confidence attributed to it.

Each *bpa* is such that  $m : 2^\theta \rightarrow [0, 1]$ , where  $\theta$  indicates the so called *frame of discernment*. Usually, the frame of discernment  $\theta$  consists of  $M$  mutually exclusive and exhaustive hypotheses. In our case, we only consider two hypotheses (classes), namely *Normal* and *Attack*; hence, the frame of discernment is  $\theta = \{\{Normal\}, \{Attack\}\}$  and  $2^\theta = \{\{Normal\}, \{Attack\}, \{Normal, Attack\}\}$ .

If we have  $n$  base classifiers, once *bpa*'s have been assigned to them, we can use the Dempster's combination rule [8] for fusing evidences coming from such classifiers. Note that the combination of several *bpa*'s by means of the Dempster rule is still a *bpa*, so we need a criterion for obtaining a *crisp* classification of each packet from the overall *bpa*. To this aim, we defined an index which allows us to associate a single *crisp* label to each packet. This *reliability index* *RI* tells us how much difference is there between the degree of belief the system has in stating each of the possible hypotheses. Such a reliability index should preferably have a direct dependency on the difference between the degrees of belief in each of the two simple hypotheses, and an inverse dependency on the degree of belief in the composite hypothesis, which explicitly measures the uncertainty associated to each decision. So, we defined *RI* as (see also [6] for further details):

$$RI = \frac{m(\{Attack\}) - m(\{Normal\})}{1 + m(\{Normal, Attack\})} \quad (1)$$

In order to reject unreliably labeled packets, the value of *RI* can be compared with a threshold  $\tau$ . If it falls outside the range  $[-\tau, \tau]$ , then the packet can be

<sup>1</sup> <http://www.snort.org>

classified; otherwise, the packet will be rejected. Note that the higher the threshold, the more reliable the classification of each packet, the higher the number of rejected packets.

## 2.2 Assigning bpa's to B-IDS

The D-S combination rule is usually referred to as a technique which requires training at fusion level [10]. In fact, the *bpa*'s are usually assigned according to the so-called *confusion matrix*, which describes the performance of a specific classifier on a given training set [13]. In our case, we don't have any training data available, hence we assign a *bpa* to each of them according to the typical performance of the detection technique it implements and according to the category (i.e., anomaly vs. misuse detection, unsupervised vs. supervised, etc.) it belongs to.

In particular, signature-based misuse detectors are built to correctly detect the malicious behaviors described by the known signatures. On the other hand, the absence of alerts by a signature-based misuse detector is not necessarily related to the absence of any anomalies. So, if we denote with  $m_{A|A}^{sb}$  the value  $m(\{Attack\})$  attributed to the evidence of an attack when a signature-based IDS attributes a packet to the *Attack* class, and with  $m_{N|N}^{sb}$  the value  $m(\{Normal\})$  attributed to the evidence of normal behavior when the same IDS does not declare the presence of an attack, we must have:

$$m_{A|A}^{sb} > m_{N|N}^{sb} \quad (2)$$

Anomaly based signature detection systems, instead, have a dual behavior with respect to misuse detectors. Hence, denoting with *ad* an anomaly-based IDS, we can assume that:

$$m_{N|N}^{ad} > m_{A|A}^{ad} \quad m_{A|A}^{ad} < m_{A|A}^{sb} \quad (3)$$

In section 4 we will show how to assign reasonable value to the *bpa* of each chosen B-IDS, according to the above described criteria.

## 3 Integrating a Supervised-IDS (S-IDS)

As stated in the introduction, the labeling performed by means of the system described in the previous Section can be used for training supervised PR-based IDS. Such IDS should perform better in real environment with respect to unsupervised IDS. As a matter of fact, they have a better generalization power, thus allowing the detection of novel anomalous events, though keeping good performance in detecting well known attacks. This section is devoted to show how an S-IDS can be integrated in the described architecture.

First of all, let us recall that an S-IDS obviously needs a training phase. In our case, for the training we'll use the results of the classification performed by the B-IDS. In the previous Section we introduced the threshold  $\tau$  on the value of *RI*, which determines the degree of reliability of packet labeling. Different

threshold values give rise to sub-datasets consisting of different fractions of the whole dataset, which can be considered reliable enough for training. To this extent, we have to understand how to select the optimal value for  $\tau$ , which allows us to obtain the best trade-off between reliability of the labeling and representativeness of the portion of the dataset selected for training.

Secondly, we must be able to understand how good the training of the S-IDS is at each stage, i.e how to measure the performance of an S-IDS, since we do not know the true labels, but only the ones provided by the ensemble of B-IDS. Furthermore, as we want to integrate an S-IDS in the whole architecture, we need to find a way to define a degree of confidence for such S-IDS, and then the corresponding *bpa* for it.

By using the new assigned labels, then, it is possible to re-train the S-IDS and to combine it again with the ensemble of B-IDS, in order to further improve the labeling. The problem here is to understand when to stop such an iterative process.

The main steps of the procedure for adding an S-IDS to the proposed architecture, aimed at obtaining a more reliable labeling, can be then summarized as follows:

- Train the S-IDS using a certain value for the reliability threshold  $\tau$
- Evaluate the performance of the trained S-IDS (see Section 3.1)
- Calculate the *bpa* for the trained S-IDS (see Section 3.2)
- Combine both B-IDS and S-IDS and evaluate their performance in terms of *Estimated Error Rate* (EER) (see Section 3.3)

These steps should be executed for each possible threshold value, in order to select the one which allows to obtain the best performance. Moreover, the insertion process of an S-IDS in the architecture can be re-iterated by using the newly computed labels. In this case, nothing ensures that the threshold value selected at the previous iteration (*level*) will lead to the best possible system performance. In practice, we have to perform a Breadth-First Search in a tree, by searching the node which gives rise to the best system configuration.

An exhaustive search of the optimal solution would be NP-hard, hence we decided to pursue a suboptimal, though computationally affordable, search strategy, as described in Section 3.3.

### 3.1 Evaluating S-IDS Performance

The problem of assigning a *bpa* to each S-IDS is strictly related to defining the extent to which we rely on the outcomes of such classifiers, whenever they are reporting a specific event. Usually, the *confusion matrix* is a good means of evaluating how good a classifier performs on a given dataset. In our case, however, we do not know the true class of a packet, but we have only the guessed class provided by the combination of the outputs of a set of IDSs. Moreover, due to the presence of the reliability index *RI*, all the packets of our dataset are *soft* labeled; hence, we have to redefine the confusion matrix. In order to



do that, each instance  $I$  of the data to be classified can be associated to an element of the set  $\{Positive, Negative\}$ , or  $\{P, N\}$  for the sake of brevity. In order to distinguish between the actual class labels and those assigned by the classifier, the former will be indicated as  $P$  and  $N$ , while the latter will be indicated as  $\hat{P}$  and  $\hat{N}$ . In our case we have a set of packets, labeled using the B-IDS part of our proposed architecture; each packet has an associated reliability index, which we denote with  $RI$ . The closer to 1  $RI$  is, the more confident we are in assigning the packet to the *Attack* class. On the other hand, the closer the index is to  $-1$ , the more we are confident in saying that the packet can be assigned to the *Normal* class. First of all, for each packet, let us assume that it belongs to the  $P$  class if  $RI > 0$  and to the  $N$  class otherwise. The same packet will be attributed to  $\hat{P}$  if the considered S-IDS recognizes it as an *Attack*, or to  $\hat{N}$  otherwise. So, we can denote with  $TP$  (true positives) the total number of packets assigned to  $\hat{P}$  by the S-IDS under exam, among all the packets whose class is  $P$ . In a similar way we can also define  $TN$ ,  $FN$  and  $FP$ .

Then, let us recall that our aim is to evaluate the S-IDS performance, in order to assign a *bpa* to it. We want to have classifiers able to correctly classify all the packets characterized by a value of  $RI$  close to either  $-1$  or  $1$ . On the other hand, if the system is not able to correctly classify packets whose value of  $RI$  is close to  $0$ , it is acceptable that they are explicitly *rejected*. In order to assign different weights according to the value of  $RI$ , we introduce the concept of a *Weighted Confusion Matrix*. Let  $\mathbf{RI} = (RI_1, RI_2, \dots, RI_m)$  be the vector of all the different reliability values associated to the packets. We define in the usual way [5] the *true positive* ( $tp$ ), the *false positive* ( $fp$ ), the *true negative* ( $tn$ ) and the *false negative* ( $fn$ ) rates:

$$tp = \frac{TP}{P} \qquad fp = \frac{FP}{N} \qquad tn = \frac{TN}{N} \qquad fn = \frac{FN}{P}$$

As an example, we show how to calculate the *weighted true positive rate*, say  $tp_{\mathbf{RI}}$ , which is function of the vector  $\mathbf{RI}$ . The same considerations can be easily extended to  $fp$ ,  $tn$  and  $fn$ . Let's first consider how we can rewrite  $tp$ :

$$tp = \frac{TP}{P} = \frac{\sum_{i=1}^m TP_{RI_i}}{P} = \frac{\sum_{i=1}^m \frac{TP_{RI_i}}{P_i} P_i}{P} = \frac{\sum_{i=1}^m (tp_{RI_i}) P_i}{P}$$

where  $TP_{RI_i}$  represents the true positives having a reliability index equal to  $RI_i$  and  $P_i$  is the number of the packets whose reliability is  $RI_i$ .

Now, by substituting  $tp_{RI_i}$  with  $tp_i = w(RI_i) \cdot tp_{RI_i}$  we obtain a true positive rate weighted by a function of the reliability index, namely  $w(RI_i)$ . So, we finally arrive at the definition of  $tp_{\mathbf{RI}}$ :

$$tp_{\mathbf{RI}} = \frac{(w(RI_1))(tp_{RI_1})P_1 + \dots + (w(RI_m))(tp_{RI_m})P_m}{P} = \frac{TP_{\mathbf{RI}}}{P}$$

The choice of the weight function can either slow down or speed up the convergence of the training algorithm for an S-IDS. The weight is supposed to

increase together with the value of the reliability index. Hence, by recalling that  $-1 \leq RI_i \leq 1$ , it must be:

$$RI_i = 0 \Rightarrow w(RI_i) = 0 \qquad RI_i = \pm 1 \Rightarrow w(RI_i) = 1$$

Since for a reliably labeled packet  $i$  it is  $RI_i = \pm 1$ , such a piece of information is worth the maximum relative weight possible. On the other hand, for packets labeled with very low reliability we know that  $RI_i \approx 0$ , hence such a contribution is supposed not to influence the final decision.

It is reasonable, given these considerations, to choose for our purposes:

$$w(RI_i) = |RI_i|.$$

### 3.2 From Weighted Confusion Matrix to bpa

The method we adopt to evaluate the *bpa* starting from the modified version of the confusion matrix relies on three parameters: the *Recognition Rate* ( $r$ ), the *Error Rate* ( $e$ ) and the *Rejection Rate* with respect to the possible classes (in our application, just *Normal* and *Attack*). Such parameters can be obtained from the confusion matrix of a classifier. In particular, as shown in [13], given a generic class  $C$ , the corresponding *bpa*, and the confusion matrix, will be:

$$m(C) = r \qquad m(\neg C) = e \qquad m(\Theta) = 1 - r - e$$

With respect to a given class  $C$ ,  $r$  is the percentage of samples assigned to class  $C$  by the classifier, and actually belonging to such a class;  $e$  is the percentage of samples belonging to class  $C$  but assigned by the classifier to some other class. By rejection rate we mean the percentage of samples which are not assigned to any class; however, the S-IDS we are using at the moment don't implement rejection explicitly. Yet, by observing the values of the weighted confusion matrix, we can see that, due to the weighting operation:

$$tp_{RI} + fn_{RI} \leq 1 \qquad tn_{RI} + fp_{RI} \leq 1$$

That stated, the *bpa* assigned to the S-IDS after its training will be:

- when the S-IDS attributes a packet to the *Attack* class:

$$\begin{aligned} m(\{Attack\}) &= tp_{RI} & m(\{Normal\}) &= fn_{RI} \\ m(\{Normal, Attack\}) &= 1 - tp_{RI} - fn_{RI} \end{aligned}$$

- when the S-IDS attributes a packet to the *Normal* class:

$$\begin{aligned} m(\{Attack\}) &= fp_{RI} & m(\{Normal\}) &= tn_{RI} \\ m(\{Normal, Attack\}) &= 1 - tn_{RI} - fp_{RI} \end{aligned}$$

The values of the weighted confusion matrix are those obtained after training the S-IDS with the labeled packets computed by the ensemble of B-IDS.

### 3.3 Terminating the S-IDS Training Process

After the first training iteration, we obtain new labels for the packets. By combining the outcome of each S-IDS with the outcome the B-IDS ensemble, we obtain a modified and hopefully more accurate labeling for the dataset. After the first step, we evaluate the *bpa* for each of the S-IDS again, in an iterative fashion. In order to define a termination criterion, we introduce the concept of *Estimated Error Rate (EER)*:

$$EER = \frac{FP_{\mathbf{RI}} + FN_{\mathbf{RI}}}{N + P}$$

where  $N + P$  is the total number of packets and  $FP_{\mathbf{RI}} = fp_{\mathbf{RI}} \cdot N$ , while  $FN_{\mathbf{RI}} = fn_{\mathbf{RI}} \cdot P$ . In order to iterate the training phase, we can decide to use, for each S-IDS, the configuration associated to the lowest *EER* (say  $EER^{best}$ ). Since this approach can lead to suboptimal results, we also consider some S-IDS training configurations which lead to higher *EER*, since during the following iterations they might lead to better overall results. In particular, we keep all the training configurations associated to a value for *EER* such that:

$$\frac{EER^{best}}{EER} \geq 0.95$$

At each iteration we use an exhaustive procedure, by training each S-IDS with all the dataset obtained by rejecting the packets characterized by reliability values below a given threshold  $\tau$ .

In order to terminate the iterative process, we evaluate the condition:

$$EER_k^{best} \leq \delta$$

where:

$EER_k^{best}$  is the estimated error rate for each S-IDS, with respect to the  $k - th$  iteration. More precisely, it is the minimum error rate among all the chosen configurations for each training iteration.

$\delta$  is a threshold value, chosen according to the specific application and the required reliability level.

## 4 Experimental Results

In order to test the proposed approach, we used one day of the DARPA 99 traces (more than 1e6 packets; among them about 1,000 are attacks). Even if the collection of these traffic traces has been criticized, they still represent the most widely used public dataset, and then can be considered a significant benchmark for testing our architecture. To extract representative features from the raw traffic traces, as required by PR-based systems, we used the preprocessor plug-in for Snort<sup>TM</sup> described in [4] and freely available on SourceForge<sup>2</sup>.

<sup>2</sup> <http://s-predator.sourceforge.net>

As B-IDS, we chose a signature-based IDS, namely Snort<sup>TM</sup>, and two PR-based anomaly detectors, respectively a Rival Penalized Competitive Learning (RPCL) network and a one-class Support Vector Machine (SVM) [1]. The RPCL has only two neurons; after the unsupervised learning phase, we associate to the *Normal* class the neuron which won the competition the more number of times for the packets under exam; by doing this, we implicitly assume that *normal* packets are in a much higher number than *attack* packets. It is worth observing that, since Snort<sup>TM</sup> works at packet level, we need to use it as a classifier for unprocessed raw tcpdump traffic. The performance of Snort<sup>TM</sup> and of the neural based classifiers on the chosen dataset are reported in table 1.

**Table 1.** Performance obtained by the chosen B-IDS

	FA Rate	MD Rate	Err Rate
<b>Snort<sup>TM</sup></b>	0.129%	21.789%	0.164%
<b>RPCL</b>	1.477%	21.997%	1.510%
<b>SVM</b>	0.924%	74.571%	1.044%

Now, we are required to model the degree of belief associated to each base classifier reporting each of the observed events. As stated before, we do it by using the basic probability assignment. First of all, let's start by assigning a *bpa* to Snort<sup>TM</sup>. As shown in table 2, we assume that Snort<sup>TM</sup> is very reliable when reporting malicious activities. The probability that an attack actually occurs when Snort<sup>TM</sup> reports it, is very high, due to its signature-based nature. According to that, we assigned the value 0.9 to  $m(\{Attack\})$  when Snort<sup>TM</sup> raises an alert. Conversely, aware of Snort<sup>TM</sup>'s low probability of detecting either a novel

**Table 2.** *bpa*'s assigned to Snort<sup>TM</sup>

	$m(\{N\})$	$m(\{A\})$	$m(\{N,A\})$
<b>A</b>	0	0.9	0.1
<b>N</b>	0.6	0	0.4

**Table 3.** *bpa*'s assigned to RPCL and SVM

	$m(\{N\})$	$m(\{A\})$	$m(\{N,A\})$
<b>A</b>	0	0.5	0.5
<b>N</b>	0.6	0	0.4

or modified attack, we don't really trust it too much when reporting normal activity. Similarly to the case of Snort<sup>TM</sup>, we considered the typical behavior of the neural-based classifiers we chose, in order to decide when and how much to trust each of them. Such classifiers, due to their unsupervised nature, are not supposed to perform extremely well on any class of traffic, since they're not meant to be too specialized. Yet, on the other hand, they may also be able to detect novel attacks, thus compensating the lack of such ability for Snort<sup>TM</sup> and similar signature based B-IDS. In table 3 we observe the *bpa*'s assigned to it. By using such *bpa*'s, and combining the results for the three considered B-IDS, we are able to obtain a preliminary labeling of the raw traffic. To this regard,

we verified that there are no differences in such a labeling if the above reported *bpa*'s vary within a 5% range and the constraints of equations (2)-(3) still hold.

By varying the value of the reliability threshold and comparing it to the indicator defined in eq. (1), we finally obtain the error rates described in table 4. Here, *False Alarm Rate* and *Rejection Rate* are meant in the canonical way. When computing the *Missed Detection Rate*, we consider all the attack packets which were either misclassified as normal, or rejected. In other words, if an attack packet is rejected, we consider it as missed. When evaluating the *Error Rate*, instead, we consider all the packets which are misclassified, regardless the number of rejections.

It can be noted in table 4 that if we consider only packets with an *RI* index greater than 0.3, we are able to obtain a better performance with respect the one exhibited by the best B-IDS. In this case, anyway, about 3,000 packets are not classified.

**Table 4.** Performance obtained by combining the chosen B-IDS as a function of the threshold  $\tau$

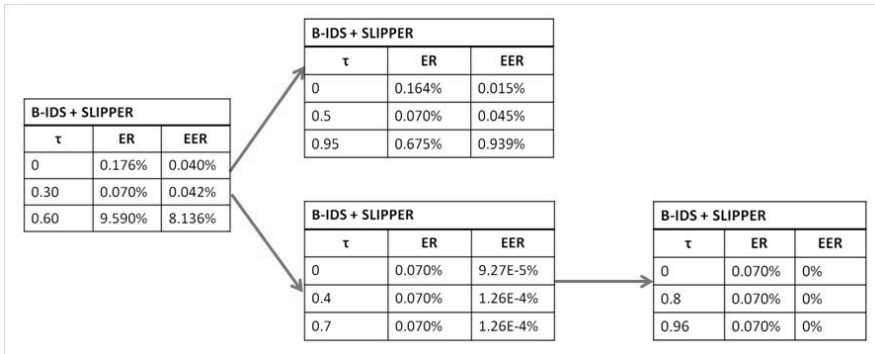
$\tau$	FA Rate	MD Rate	Err Rate	Rej Rate
0.0	0.278%	21.789%	0.313%	0%
0.3	0.027%	21.945%	0.063%	0.251%
0.6	0.015%	74.623%	0.051%	2.404%
0.9	0.015%	74.623%	0.015%	99.943%

#### 4.1 Training and Integrating an S-IDS

In this section we will describe the *evolution* of an S-IDS. Here we recall that such classifiers always need a training phase performed on pre-classified traffic. In this case, we will first use the traffic labeled by the ensemble of B-IDS for training, by using certain reliability thresholds.

In the following, we will show an example of S-IDS integration with the rest of the system. As S-IDS, we have chosen SLIPPER [2], a rule-based learning system that creates a rule set by iteratively boosting a greedy rule-builder. To such extent, we'll describe how the whole procedure applies to the case of SLIPPER. In figure 1 we show how different configurations for the threshold are chosen, according to the value of *EER*, until the termination condition is met. In particular, we considered a  $\delta$  value equal to  $1e - 5$ , which is reached at the third training iteration. In the figure are also depicted the *actual* error rate *ER* and the estimated error rate *EER*. The former is evaluated on the reference dataset, by using the actual labels.

In the proposed example, during the first iteration, we choose two values for the threshold associated to very similar values for *EER*, whose ratio is over the 0.95 threshold fixed in Section 3.3. The results provided by the proposed optimization procedure will actually lead to the best possible performance (i.e. the one obtained if the whole tree would be searched).



**Fig. 1.** Results obtained by the proposed architecture at different training iterations. The procedure described in Section 3 is used for searching the configuration leading to the lowest *EER*

At this point, we want to show how, through the proposed labeling scheme, it is possible to obtain better training results than the ones obtained by the combination of the B-IDS alone. This would both justify the increased complexity of the architecture resulting in increased requested computational power, and the need for multiple iterations over the same data, in order to improve the knowledge of the whole system. In order to do that, we refer to table 5, where we show the performance obtained by SLIPPER when it’s trained by using the labels provided by the ensemble of B-IDS and the proposed architecture (first two rows). For the sake of comparison, also the results obtained by training SLIPPER with different fractions of the dataset with the actual labels are reported (rows 3–6).

In this case, the difference with respect to the ideal training set (i.e., the whole database with actual labels) in terms of error rate turned out surely acceptable (only 0.03% with a recognition rate of 99.9%), thus confirming our claims.

**Table 5.** SLIPPER performance as a function of the dataset used for training

		FA Rate	MD Rate	Err Rate
labels provided by the bank of B-IDS	$\tau = 0$	0.141%	21.997%	0.176%
labels provided by the proposed architecture	$\tau = 0$	0.035%	21.945%	0.070%
actual labels	10%	0.045%	88.820%	0.190%
	20%	0.025%	79.407%	0.154%
	30%	0.004%	19.345%	0.036%
	100%	0.013%	12.117%	0.032%

Moreover, it has to be pointed out that labels provided by our approach allow us to train a classifier that performs better than the one trained with the 20% of the whole database with actual labels, which anyhow exhibits an unacceptable missed detection rate.

We have also conducted a statistical significance analysis with the  $t$ -test, as suggested in [9]. Within a significance threshold of 1%, the difference between results obtained by our approach and those obtained by considering the 20% of the whole database with actual labels resulted significant. This is a very good result, since such a dataset is composed by over 230,000 samples that have been labeled by hand, with a significant waste of time with respect to an automatic labeling.

## 5 Conclusions

In this paper we deal with the problem of collecting labeled packets, in order to usefully train intrusion detection systems. We describe a self-training architecture based on the Dempster-Shafer theory which, by analyzing raw packets, is able to assign labels to them. It is also able to point out how reliable each of the labels is, thus allowing to choose only the packets which satisfy a certain level of confidence. By training a supervised algorithm with data labeled in this way, we show that it is possible to obtain results which are similar to those obtained by using the actual class of all the packets.

As future developments, we want to test the whole architecture on other traffic traces. Furthermore, we think it'd be an interesting development to study the impact of the choice of different weighting functions such as the ones described in section 3.1. Of course, such function might impact both the accuracy of the labeling and the computational resource demand involved in the whole process.

**Acknowledgement.** This work has been partially supported by the Ministero dell'Università e della Ricerca (MiUR) in the framework of the *RECIPE* Project, and by the Regione Campania, in the framework of the *SCI-TrADe* Project.

## References

1. C.-C. Chang, C.-J. Lin, LIBSVM: A Library for Support Vector Machines, 2001.
2. W.W. Cohen, Y. Singer. Simple, Fast, and Effective Rule Learner. In *Proceedings of the 16th National Conf. on Artificial Intelligence and 11th Conf. on Innovative Applications of Artificial Intelligence*, Orlando (FL), USA, pp. 335-342, July 18-22, 1999.
3. C. Elkan, Results of the KDD99 classifier learning, *ACM SIGKDD Explorations* 1, pp. 63-64, 2000.
4. M. Esposito, C. Mazzariello, F. Oliviero, S. P. Romano, C. Sansone, Real Time Detection of Novel Attack by Means of Data Mining Techniques. In: C.-S. Chen, J. Filipe, I. Seruca, J. Cordeiro (Eds.), *Enterprise Information Systems VII*, Springer-Verlag, pp. 197-204, 2006.

5. T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters* vol. 27, pp. 861-874, 2006.
6. F. Gargiulo, C. Mazzariello, C. Sansone, Information Fusion techniques for reliably Training Intrusion Detection System. In Proc. of *IWAPR 2007*, Springer Verlag, 2007 (in press).
7. 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.
8. J. Gordon, E.H. Shortliffe, The Dempster-Shafer Theory of Evidence, in B.G. Buchanan and E.H. Shortliffe (Eds.), *Rule-Based Expert Systems*, Addison-Wesley, pp. 272-292, 1984.
9. L.I. Kuncheva, Switching Between Selection and Fusion in Combining Classifiers: An Experiment. In *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, no. 2, 2002.
10. L.I. Kuncheva, J.C. Bezdek, R.P.W. Duin, Decision templates for multiple classifier fusion: an experimental comparison. In *Pattern Recognition*, vol. 34, pp. 299-314. 2001.
11. P. Laskov, P. Daussel, C. Schafer, K. Rieck, Learning intrusion detection: supervised or unsupervised?, in F. Roli and S. Vitulano (Eds.), *Lecture Notes in Computer Science*, vol. 3617, Springer-Verlag, Berlin, pp. 50-57, 2005.
12. M.V. Mahoney, P.K. Chan, An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection, in G. Vigna, E. Jonsson, C. Kruegel (Eds.), *LNCS vol. 2820, Proceedings of RAID 2003*, pp. 220-238, 2003.
13. L. Xu, A. Krzyzak, C.Y. Suen, Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. *IEEE Trans. on SMC*, Vol.22, pp. 418-435, 1992.
14. S. Zanero, Analyzing tcp traffic patterns using self organizing maps, in F. Roli and S. Vitulano (Eds.), *Lecture Notes in Computer Science*, vol. 3617, Springer-Verlag, Berlin, pp. 8390, 2005.
15. C. Zhang, J. Jiang, M. Kamel, Intrusion detection using hierarchical neural networks. *Pattern Recognition Letters* 26(6), pp. 779-791, 2005.
16. X. Zhu, Semi-Supervised Learning Literature Survey, *Computer Sciences, University of Wisconsin-Madison* 1530, [http://www.cs.wisc.edu/~jerryzhu/pub/ssl\\_survey.pdf](http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf), 2005.



---

# A Case-Base Reasoning System for Predicting the Economic Situation of Enterprises – Tacit Knowledge Capture Process (Externalization)

Jan Andreasik

College of Management and Public Administration in Zamość, ul. Akademicka 4,  
22-400 Zamość, Poland  
jandreasik@spp.org.pl

**Summary.** The Case-Based Reasoning methodology is used to develop Knowledge Management Systems (KMS). First, this paper introduces various case-based reasoning cycles. Next, the focus is on defining procedures for case indexing. The procedures are based on multiple-criteria methods supporting a decision-making process, such as EUCLID and ELECTRE TRI. The indexing procedures form also a knowledge transformation cycle (from implicit to explicit knowledge) known as externalization. The author developed a subsystem, which supports the implementation of this cycle. The UML diagrams show the structure of the SOK-P1 subsystem. The author developed an ontology of enterprise assessment. The subsystem presented uses the taxonomies of the enterprise potential and business risks, as well as thesauruses, which are a set of suggestions for assessors.

## 1 Introduction

The basic structure of an intelligent case-based system in the Case-Based Reasoning methodology (CBR) is a CBR cycle. A. Aamodt and E. Plaza [1] were the first who defined the CBR cycle (the four REs) consisting of the following four processes:

1. RETRIEVE - the process of selecting the most similar cases to a given one from a case base.
2. REUSE - the process of adapting the solution developed in selected cases to the problem presented.
3. REVISE - the process of evaluating and verifying the solution.
4. RETAIN - the process of updating the case base with unprecedented cases.

T. Reinartz, I. Iglezakis and T. Roth-Berghofer [2] divided the CBR cycle into two phases:

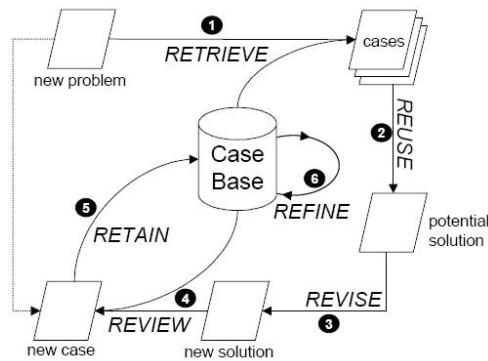
- I. Reasoning process implementation phase (Application Phase).
- II. Base creation phase (Maintenance Phase).

There are three processes included into the first phase: RETRIEVE - selection process, REUSE - adaptation process, REVISE - evaluation and correction process. The second phase also consists of three processes: RETAIN - case-base update process and two new ones:

5. REVIEW - verification process.
6. RESTORE - reconstruction process.

I. Watson [3] presented the CBR methodology as a framework for designing knowledge management systems. Six processes are mentioned in their proposal of a reasoning cycle:

1. RETRIEVE - selection of similar cases.
2. REUSE - adaptation of the solution.
3. REVISE - evaluation and correction of the solution
4. REVIEW - verification.
5. RETAIN - entering unprecedented cases to the database.
6. REFINE - capture of tacit knowledge (indexing and feature weight estimation)



**Fig. 1.** The case-based reasoning cycle (CBR) as per [3]

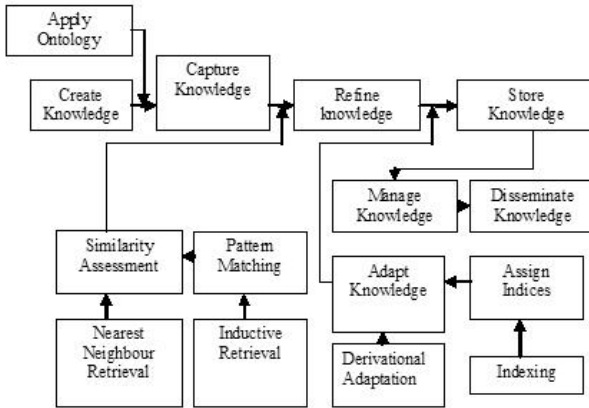
The first step in designing intelligent systems with a case-base is to define the ontology of cases related to a given subject. The next step is to define the approach to case representation and the structure of the case-base.

To ensure a proper definition of the case-base structure a list of indices should be provided and the procedures explaining the index values should be developed. These tasks are performed by the knowledge refinement process (REFINE) [3].

Following Aamodt's and Plaza's definition of the case-based reasoning cycle [1] the researchers were searching for a method to integrate the CBR cycle and the knowledge management cycle. A proposal of such integration, introduced by S.H. Kang and S.K. Lau in [4], is presented in the Table 1 and on the diagram below (Fig. 2).

**Table 1.** The CBR cycle and the Knowledge Management (KM) cycle

CBR cycle	KM (Knowledge Management) cycle
Retrieve - selection	Knowledge creation, capture
Reuse - adaptation	Knowledge refinement
Revise - evaluation and verification	Knowledge refinement
Retain - update	Knowledge storage, management, dissemination



**Fig. 2.** Integration of the CBR and KM cycles according to [4]

Extensions of the classic case-based reasoning cycle result from two main problems analyzed in the CBR methodology:

1. Knowledge representation.
2. Case-base development.

The development of an intelligent CBR-based system requires solutions for representation of tacit knowledge, which may be captured from databases.

I. Watson [4] presented a CBR cycle based on the knowledge management cycle (Fig. 3). This diagram is suitable for systems capturing knowledge from experimental databases. Such systems are used in the processes of technical or medical diagnosis, where it is possible to obtain data using a measuring equipment.

Another concept of linking the Knowledge Based Systems to the CBR methodology was presented by T.R. Roth-Berghofer and J. Cassens [5]. Their proposal is to associate the explanation processes in the CBR systems with so-called “knowledge containers”. Four knowledge containers were identified:

1. “Vocabulary” container.
2. “Case-base” container.



- linear evaluation scale within a specified range<sup>1</sup>,
- verbal scale<sup>2</sup>,
- preference interrelationships [6].

The fundamental issue to be addressed in beginning of decision support system design is the description of the variants [6]. The assessment of an enterprise situation requires a specific formal model conforming to the strategy of the enterprise. In this paper the author adopted a useful formal model first proposed by M. Tavana and S. Banerjee [7] and then modified by M. Tavana [8]:

$$S^m = \sum_{i=1}^I W_i \left( \sum_{j=1}^J F_{ij} \beta_{ij} \left( P_{ij}^m \right) \right) \tag{1}$$

where:

- $P_{ij}^m$ - assessment of benefit or risk,
- $W_i$ - weight of the i-th group of criteria,
- $F_{ij}$ - weight of the j-th criterion in the i-th group,
- $\beta_{ij} \in \{-1, +1\}$ ,
- $0 \leq P_{ij}^m \leq 1$ ,
- $0 \leq W_i \leq 1$ ,
- $\sum_{i=1}^I W_i = 1$ ,
- $0 \leq F_{ij} \leq 1$ ,
- $\sum_{j=1}^J F_{ij} = 1$ .

The S case (variant) is assessed by means of two opposite groups of criteria: the benefits criteria ( $\beta = +1$ ) and the risk criteria ( $\beta = -1$ ).

There are subgroups identified in each group of criteria. Each subgroup is characterized by a weight W. The expert makes an assessment of the S variant (case) according to each criterion by providing his/her subjective assessment P and the weight of the criterion F. The W weight for a group of criteria is estimated using the AHP method [11].

---

<sup>1</sup> The most commonly used scale is that developed by T.L. Saaty, in the range of [1/9, 9]. Evaluation of different scales is provided by E. Triantaphyllou in [9].

<sup>2</sup> Formalization of verbal scales is discussed by Tavana, e.g. in [10]. The authors accepted the following probability thresholds for particular phrases:

- impossible (benefit, risk)- 00,
- small possibility (benefit, risk)- 0.10,
- small chance (benefit, risk)- 0.20,
- somewhat doubtful (benefit, risk)- 0.30,
- possible (benefit, risk)- 0.40,
- toss-up (benefit, risk)- 0.50,
- somewhat likely (benefit, risk)- 0.60,
- likely (benefit, risk)- 0.70,
- very likely (benefit, risk)- 0.80,
- quite certain (benefit, risk)- 0.90,
- certain (benefit, risk)- 1.00

M. Tavana [12] developed an EUCLID model to classify strategy variants. In this model two indices are calculated: the benefit index  $B^m$  and the risk index  $R^m$ .

$$B^m = \sum_{i=1}^3 W_{b_i} \left( \sum_{j=1}^{J_{b_i}} F_{b_{ij}} \left[ \left( P_{b_{ij}}^* - P_{b_{ij}}^m \right)^2 \right] \right) \tag{2}$$

$$R^m = \sum_{i=1}^3 W_{r_i} \left( \sum_{j=1}^{J_{r_i}} F_{r_{ij}} \left[ \left( P_{r_{ij}}^* - P_{r_{ij}}^m \right)^2 \right] \right) \tag{3}$$

where:

- $P_{r_{ij}}^*$  - minimum value of the assessment of the i-th risk, j-th component,
- $P_{b_{ij}}^*$  - maximum value of the assessment of the i-th benefit, j-th component.

Therefore, the formulas (2) and (3) describe the principle of indexation through aggregating experts' opinions. Comparisons are made to extreme assessments, i.e. to the maximum benefit and the minimum risk assessments.

By adopting the model above the author developed an expert system for enterprise assessment.

A comprehensive assessment of the enterprise situation is made by the experts. The assess the enterprise strategy in five potential-risk perspectives:

1. B1-R1 perspective: Capital Potential – Risk of satisfying the demand for capital.
2. B2-R2 perspective: Innovation and Investment Potential – Risk of that the expenditures on investments and innovation are higher than planned.
3. B3-R3 perspective: Key Stakeholder Potential – Risk of increased costs of services provided to stakeholders.
4. B4-R4 perspective: Neighborhood Potential – Risk of increase of cost /decrease of revenues caused by local tendencies.
5. B5-R5 perspective: Environment Potential – Risk of increase of cost /decrease of revenues caused by global tendencies.

In each perspective an enterprise is represented by a pair of indices (B,R) calculated according to the formulas (2) and (3).

While assessing the strategic position of an enterprise in each perspective the experts determine the weights of the groups of criteria (types of potentials and risks) and the weights and assessments of each criterion in the group. The SOK-P1 systems features useful editors to enter the weights and assessments.

For each enterprise, which is subject to the experts' assessment, a matrix of results  $[y_{ij}]$  is assigned with the data for the last dozen years. The rows in the matrix are financial indicators calculated by means of various methods, whereas the columns represent the consecutive years.

These methods may be classified in the following way:

1. Methods focused on the profit analysis:
  - a) Net profit
  - b) EBIT – (earnings before income taxes)

- c) EBITDA – (earnings before interest, taxes, depreciation and amortization)
- d) NOPAT – (net operating profit after taxes)
- 2. Methods focused on the cash-flow analysis:
  - a) FCF (Free Cash Flow)
  - b) NPV (Net Present Value)
  - c) SVA (Shareholder Value Added )
  - d) CVA (Cash Value Added)
  - e) CFROI (Cash Flow Return on Investment)
- 3. Methods focused on the value-added analysis.
  - a) VA (Value Added)
  - b) EAR (Economic Added Result)
  - c) VAIC (Intellectual Coefficient Value Added)
  - d) MVA (Market Value Added)
- 4. Methods focused on the capital analysis.
  - a) CE (Capital Employed)
  - b) EVA (Economic Value Added)

A particular trajectory of the enterprise's financial situation may be created on the basis of some parameters only, e.g. income on sales, net profit, employment, EBITDA, FCF and VAIC in the last ten years.

Using a multiattribute value function [13] a standardized trajectory may be identified.

Such a trajectory will be subject to adaptation in the course of the process of predicting a future situation of an enterprise (the REUSE process – adaptation of solutions in the CBR cycle).

### 3 Case Indexation- Second Level

Among all key processes in the CBR methodology the process of explanation plays a crucial role. That is why the aggregation of experts' opinions leads to a development of indices determining the assignment of a given case to a particular class. In the multiple criteria theory of decision making B. Roy [6] defined the concept of sorting. Following this theory the ELECTRE TRI method and software were developed [14]. To utilize this method it is necessary to build a sorting machine model, which contains the definitions of profiles separating the categories. For each criterion a category limit value "g" should be defined. Since the profiles are defined as fuzzy numbers, it is required to define an indifference threshold "q", a preference threshold "p" and the veto threshold "v".

In the Tavanas case indexing model a pair of indices ( $B^m$ ,  $R^m$ ) is derived from the formulas (2) and (3).

With M cases in the database the average values are calculated:

$$\bar{B} = \sum_{m=1}^M B^m / M \quad (4)$$

$$\bar{R} = \sum_{m=1}^M R^m / M \tag{5}$$

These values divide the B-R space into four classes as depicted in the Fig. 4.

Challenge Strong Benefits and Strong Risks $B^m \leq \bar{B} \text{ ; } R^m > \bar{R}$	Desperation Weak Benefits and Strong Risks $B^m > \bar{B} \text{ ; } R^m > \bar{R}$
Exploitation Strong Benefits and Weak Risks $B^m \leq \bar{B} \text{ ; } R^m \leq \bar{R}$	Discretion Weak Benefits and Weak Risks $B^m > \bar{B} \text{ ; } R^m \leq \bar{R}$

Fig. 4. Classes in the EUCLID method [12]

As the aggregation of risk and potential assessments and weights in the EUCLID method results in two indices for each case ( $B^m, R^m$ ), a problem of a further aggregation arises to determine one index, which could serve as a criterion function.

The author of this paper proposes to define this index as a  $\theta$  angle measured between the radius-vector of the enterprise assessment and the Benefit axis (B) in a transformed set of coordinates, which axes overlap with the average values of the risks and benefits. The Fig. 5 shows a geometrical explanation of this index.

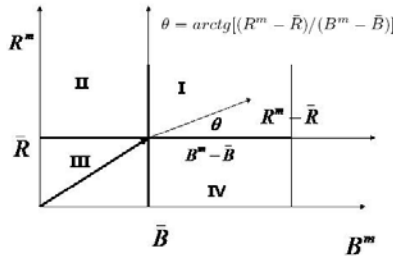


Fig. 5. Geometrical representation of the enterprise situation index  $\theta$  in the (B, R) space

The  $\theta_i$  index is calculated for each ( $B_i, R_i$ ) criterion. The indices create a set of criteria (a criterion function) for assessing the situation of the enterprise in all perspectives. The ELECTRE TRI method and software may be used to



Definition of classes and profiles for the ELECTRE TRI method

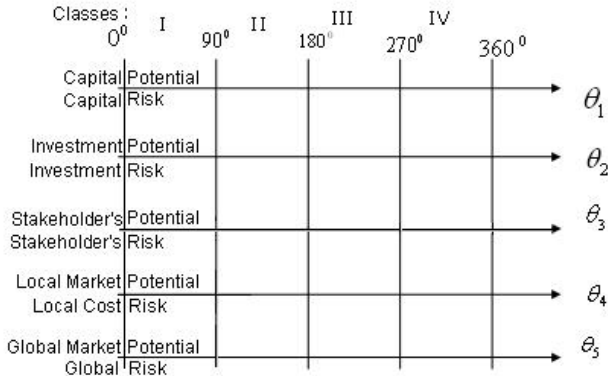


Fig. 6. Set of indices  $\theta_i$  as the criteria for the ELECTRE TRI sorting mechanism

aggregate these assessments. The Fig. 8 shows the criteria and profiles of a defined sorting machine.

Two indexation stages presented so far lead to the following definition of a case:

$$AES^3 = \langle \theta, \theta_1, d_1, \theta_2, d_2, \theta_3, d_3, \theta_4, d_4, \theta_5, d_5 \rangle \tag{6}$$

where:

$d_i$ - the length of the radius-vector of the enterprise location for,

$\theta_i$ - the angle measured between the radius-vector of the enterprise assessment and the Benefit axis (B) in the i-th perspective Bi-Pi,  $i= 1, \dots, 5$ .

### 4 Tacit Knowledge Capture Process

Apart from knowledge representation the issue of designing a tacit knowledge capture process is also important in case based systems. While diagnoses of business processes are available, e.g. from databases of financial statements, audit reports, ISO audit reports, BI (Business Intelligence) systems reports, in case of assessments of enterprise development strategy, risk of their future operations and projects we have to count on experts' and managers' knowledge.

I. Nonaka, H. Takeuchi [15] presented a system for knowledge conversion and distinguish four conversion mechanisms:

1. From tacit to tacit knowledge (socialization).
2. From tacit to explicit knowledge (externalization).
3. From explicit to explicit knowledge (combination).
4. From explicit to tacit knowledge (internalization).

<sup>3</sup> Assessment of Enterprise Situation.

This paper presents a model of conversion of tacit knowledge (possessed by experts) into the explicit one. The author developed a basic ontology of enterprise assessment. The ontology consists of:

1. Taxonomy of potentials.
2. Taxonomy of risks.
3. Thesaurus of assessment explanations.
4. Thesaurus of business performance explanations.

Fig. 9 presents the cases of using the 1st stage of tacit experts' knowledge capture (the UML [16]) in the SOK-P1 system.

In this system the process of knowledge capture relies on a cooperation of experts:

1. Assessing Expert who assesses the components of the potentials and risks.
2. Estimating Expert who estimates the weights of the assessments.
3. Defining Expert who defines the category limits.
4. Financial Analysis Expert who calculates the financial parameters of an enterprise situation trajectory.
5. Manager who manages the early warning system.

The Assessing Expert uses an editor developed on the basis of taxonomies of risks and potentials. The Expert may refer to the thesaurus of assessment explanations.

The Estimating Expert is provided with an editor, which allows entering the weights by a group of experts and then to fix the weights as average values. It is also possible to use the EXPERT CHOICE system to set the weights with the AHP method.

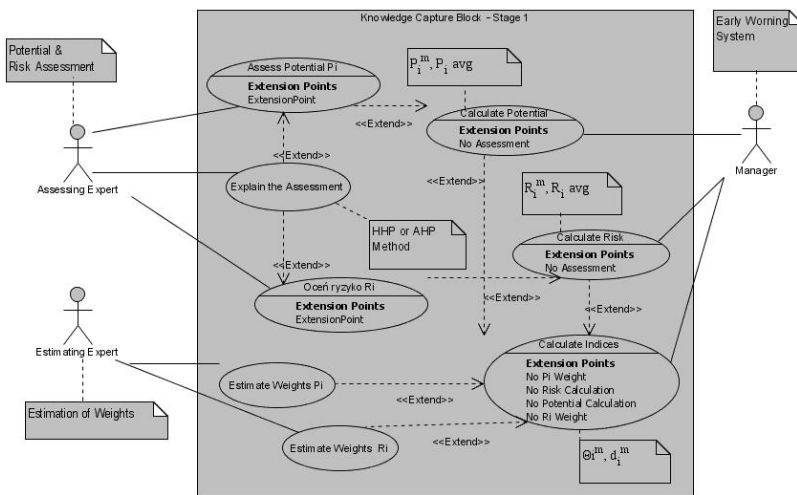
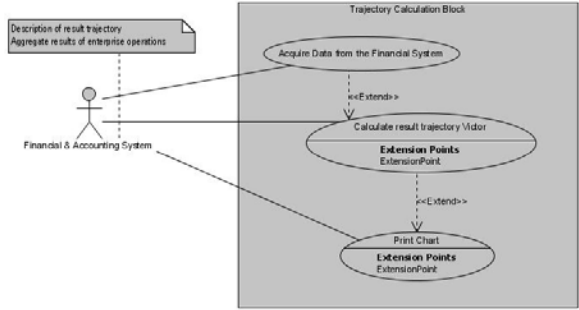
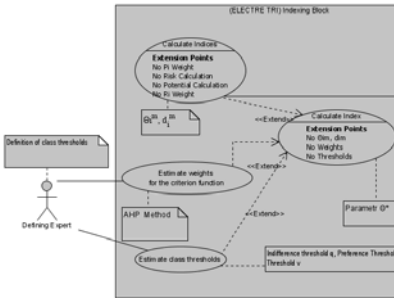


Fig. 7. Application diagram for the 1st stage of the Knowledge Capture process in the SOK-P1 system



**Fig. 8.** Application diagram for the 1st stage of the Knowledge Capture- result representation



**Fig. 9.** Application diagram for the 2nd stage of the Knowledge Capture process

The Defining Expert, who defines the category limits for the needs of the ELECTRE TRI method, sets the values of the  $p, q, v$  parameters on the basis of sorting accuracy assumed.

The Financial Analysis Expert may interpret the results calculated using the accounting system data through modifying the utility function. The Logical Decisions system may be applied here

The Manager of the early warning system may view the whole knowledge capture process, i.e. he can negotiate particular assessments, which he does not agree with or finds them doubtful, in an interactive mode. He can make the Experts provide additional justifications, which extends the thesaurus of assessment explanations.

### 5 Summary

The paper presents the first stage of development of an intelligent case-based system to predict the situation of enterprises. This stage concentrates on capturing the experts' tacit knowledge on the assessments of enterprises potential and risks in the future. The experts assess the weights of the single components

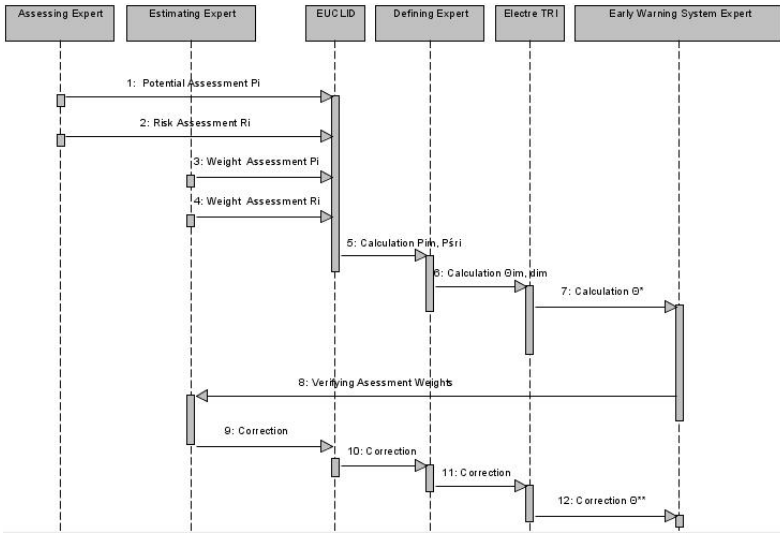


Fig. 10. Sequence diagram for the SOK-P1 system

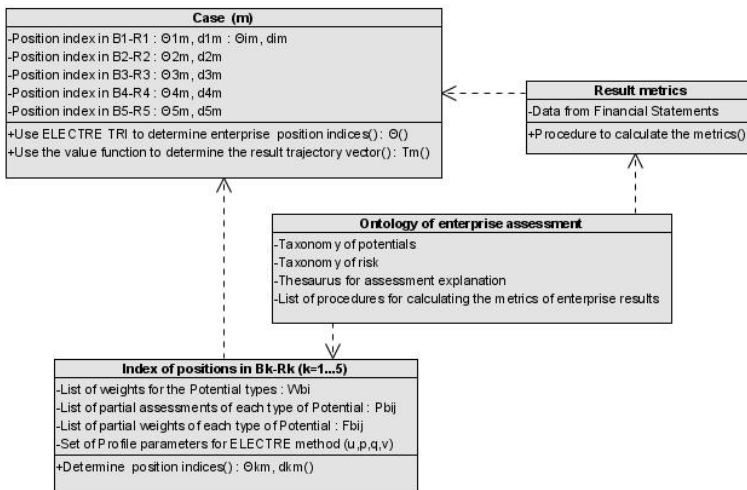


Fig. 11. General diagram of categories

of the potentials and risks. The structure of the subsystem, which supports the experts in making and verifying the assessments, is presented on UML diagrams.

The author developed the taxonomies of enterprise potentials and risks. There are thesauruses available for the experts. They include lists of possible enterprise policies, rules and trends for each single assessment. The expert may then increase the objectivity of his/her assessments. Next, the paper presents two stages

of indexation based on assessment aggregation. In the first stage the EUCLID method is used, so is the ELECTRE TRI method in the second stage. This is an original approach to determine the case indices, and thus a benchmarked position of an enterprise in the potential - risk space. This indexation concept is different from existing ones, which are based on extracting the features from a set including all possible ones using statistical methods and data mining. The approach presented in this paper is based on multiple criteria decision support methods. It is possible to develop procedures to facilitate the experts' knowledge capture process. In a knowledge conversion cycle the key process is externalization, i.e. from tacit to explicit knowledge. The paper presents also the structure of the SOK-P1 subsystem, which supports the whole process. Further works shall focus on implementation of models for particular stages of the Case-Based Reasoning cycle.

## Acknowledgement

This research was financed under the **EQUAL** project No. F0086 implemented by the **College of Public Administration and Management in Zamość**.

## References

1. Aamodt A., Plaza E. (1994) *AI Communications* (7:1):39–59.
2. Reinartz T., Iglezakis I., Roth-Berghofer T. (2001) *Computational Intelligence: Special Issue on Maintaining Case-Based Reasoning Systems* 17:214–234.
3. Watson I. (2001) Knowledge management and case-based reasoning: a perfect match? In: *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*. AAAI Press, pp.118–122.
4. Kang S.H., Lau S.K. (2002) Intelligent knowledge acquisition with case-based reasoning techniques. Technical Report University of Wollongong. NSW, Australia.
5. Roth-Berghofer T.R., Cassens J. (2005) Mapping goals and kinds of explanations to the knowledge containers of Case-Based Reasoning systems. In: *Case-Based Reasoning Research and Development*. ICCBR 2005, Springer, pp.451–464.
6. Roy B. (1985) *Méthodologie Multicritère d'Aide à la Décision*. Editions Economica, Paris.
7. Tavana M., Banerjee S. (1995) *Computers & Operations Research* 22:731–743.
8. Tavana M. (2004) *Computers & Operations Research* 31:1147–1164.
9. Triantaphyllou E. (2000) *Multi-criteria decision making methods: a comparative study*. Kluwer Academic Publishers.
10. Tavana M., Kennedy D., Mohebbi B. (1997) *Journal of Behavioral Decision Making* 10:133–150.
11. Saaty T.L. (2001) *Decision making for leaders*. RWS Publications.
12. Tavana M. (2002) *Journal of multi-criteria decision analysis* 11:75–96.
13. Kirkwood C.W. (1997) *Strategic decision making*. Dexbury Press.
14. Mousseau V., Słowiński R., Zielniewicz P. (2000) *Computers & Operations Research* 27:757–777.
15. Nonaka I., Takeuchi H. (1995) *The Knowledge-creating Company: How Japanese Companies Create the Dynamics of Innovation* (Hardcover). Oxford University Press, Inc.
16. Rhem A. J. (2006) *UML for developing knowledge management systems*. Auerbach Publications.

---

# A Structural Pattern Analysis Approach to Iris Recognition

Hugo Proença

Department of Computer Science, IT - Networks and Multimedia Group  
University of Beira Interior, Covilhã, Portugal  
hugomcp@di.ubi.pt

**Summary.** Continuous efforts have been made in searching for robust and effective iris coding methods, since Daugman's pioneering work on iris recognition was published. Proposed algorithms follow the statistical pattern recognition paradigm and encode the iris texture information through phase, zero-crossing or texture-analysis based methods. In this paper we propose an iris recognition algorithm that follows the structural (syntactic) pattern recognition paradigm, which can be advantageous essentially for the purposes of description and of the human-perception of the system's functioning. Our experiments, that were performed on two widely used iris image databases (CASIA.v3 and ICE), show that the proposed iris structure provides enough discriminating information to enable accurate biometric recognition, while maintains the advantages intrinsic to structural pattern recognition systems.

## 1 Introduction

In 1987, Flom and Safir observed the stability of iris morphology over human lifetime and estimated the probability for the existence of two similar irises at 1 in  $10^{72}$ . Since then, the use of iris-based biometrics has been increasingly encouraged by both government and private entities. The iris is accepted as one of the most reliable biometric traits: it has a random morphogenesis and, apparently, no genetic penetrance.

The published iris recognition algorithms, at least the most relevant, follow the statistical Pattern Recognition (PR) paradigm (e.g., Daugman [3], Wildes [9] and Ma *et al.* [7]). They represent patterns as sets of features and regard them as points in a  $d$ -dimensional space. In this context, effectiveness is determined by the classes separability that is given by some data projection.

The structural PR approach is based upon the view that each pattern is composed of simple sub-patterns. The most elementary sub-patterns are the *primitives*. A pattern can be expressed in term of primitives and of *relationships* between primitives. Consequently, symbolic data structures (e.g., strings, tress, and graphs) are used for pattern representation. Feature sets generated by structural PR approaches contain a varying number of features. As the relationships between the primitives are also encoded, the feature vector must include

additional components describing them or take an alternate form, such as a relational graph. Further, a pattern can be labeled as belonging to a class if its graph representation is isomorphic with some prototype graphs of the class [4].

This is intuitively appealing to biometrics because, apart classification, it simultaneously provides the description of how a given pattern is constructed from the primitives and the justification for a match/non-match produced in the comparison between two patterns. However, it is usually considered that may yields an enormous number of possibilities to be evaluated, demanding large training sets and very large computational efforts.

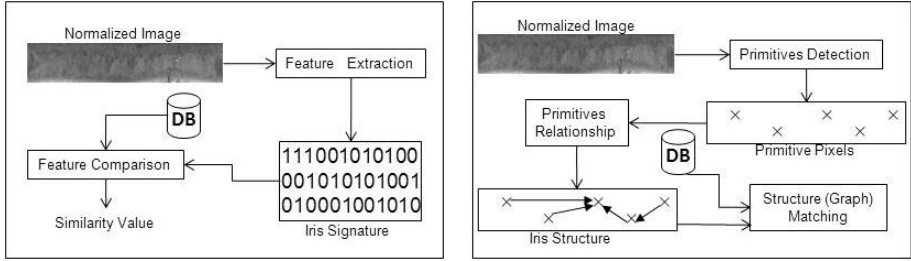
Our main goal is to show that structural-based iris recognition methods achieve very low error rates and be used for biometric purposes. Due to their intrinsic characteristics, this kind of methods facilitate the human-perception of the pattern encoding and matching, which can be useful - for instance - for forensic or security purposes. Also, the fact that our proposal does not contain any user-parameterized thresholds must be enhanced, regarding the dynamics of the imaging environments and the heterogeneity of the images that are presented to recognition systems.

The remainder of this paper is organized as follows: section 2 briefly summarizes the most cited iris recognition methods. A detailed description of the proposed method is given in section 3. Section 4 reports the experiments and results and, finally, section 5 concludes this paper.

## 2 Iris Recognition

As previously stated, the most relevant iris recognition algorithms follow the statistical PR paradigm and share the structure given in figure 1(a). After the segmentation and normalization of the iris images, the iris data is encoded through one of three major variants: phase-based methods (e.g., [3]), zero-crossing methods (e.g., [2]), and texture-analysis methods (e.g., [9]). Daugman [3] used multi-scale quadrature wavelets to extract texture phase information and obtain an iris signature with 2048 binary components. Boles and Boashash [2] computed the zero-crossing representation of a 1D wavelet at different resolutions of concentric circles. Wildes [9] proposed the characterization of the iris texture through a Laplacian pyramid with four different levels (scales). Further, the comparison between iris signatures is performed and produced a numeric dissimilarity value that determines the subjects' identity. It is common to apply different distance metrics (Hamming [3], Euclidean [5] or weighted Euclidean [7]), or methods based on signal correlation [9].

Figure 1(b) gives the block diagram of the proposed iris recognition method. According the structural PR paradigm, it starts by the detection of the primitive pixels among the iris data, followed by the detection of relationships between these primitives. This gives us a directed graph that represents the structure of the iris image and, as our experiments confirm, contains enough discriminating information to accurately distinguish between individuals.



(a) Typical stages of statistical iris recognition

(b) Stages of the proposed recognition method, according to the structural pattern recognition paradigm

Fig. 1. Statistical versus structural approaches to iris recognition

### 3 Underlying Iris Structure and Structural Matching

Due to the pupils' varying sizes and to different distances, angles and optical magnifications of the imaging frameworks, the size of the irises in the captured images will have significant variations. As described by Daugman [3], the invariance to all these factors can be obtained through the translation of the captured data into a double dimensionless polar coordinate system, in a process known as the "Daugman Rubber Sheet". To each point on the iris, regardless of its size and pupillary dilation, it is assigned a pair of real coordinates  $(r, \theta)$ , where  $r$  is on the unit interval  $[0, 1]$  and  $\theta$  is an angle in  $[0, 2\pi]$ . The remapping of the iris image from raw cartesian coordinates  $(x, y)$  to the dimensionless non concentric polar coordinate system  $(r, \theta)$  is given by  $I(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta)$ , where  $x(r, \theta)$  and  $y(r, \theta)$  are linear combinations of both the set of pupillary and limbus boundary points, detected in the iris segmentation stage.

#### 3.1 Primitives

The rationale behind our proposal is to consider as primitives the centers of the most homogeneous iris regions. These have lower average differences of intensities between neighbor pixels and are intuitively visually relevant regions. Let  $I$  be a segmented and normalized iris image. Let  $G(x, y)$  be a Gaussian kernel with dimensions  $L \times L$  that weights the differences between pixels' intensity. The function  $e(x, y) : \mathbb{N}^2 \rightarrow \mathbb{R}$  measures the distinctiveness between the pixel  $p = (x, y)$  and its neighborhood of radius  $L/2$ :

$$e(x, y) = \sum_{i=-L/2}^{L/2} \sum_{j=-L/2}^{L/2} \left[ \left| |I(x+i, y+j) - I(x, y)| \right| \times G\left(i + \frac{L}{2}, j + \frac{L}{2}\right) \right] \tag{1}$$



where  $I(x, y)$  is the image intensity at location  $(x, y)$  and  $||\cdot||$  denotes the absolute value. Based on this function, a pixel  $p = (x, y)$  is considered a "primitive" if it has minimal  $e()$  value in a neighborhood of radius  $L/2$ .

$$pr(x, y) = \begin{cases} 1 & (x, y) = \min_{\frac{L}{2}} e(x_i, y_j) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

As illustrated by figure 2, this process gives a set of primitives with variable dimension. These constitute the vertices of the graph that represents the structure of the iris image.

**3.2 Relationship Between Primitives**

In this section we describe how the relationships between primitives are established, i.e., how to find the graph's set of edges. The rationale of our method is to link neighbor vertices with high difference between their  $e()$  values. These directions correspond to regions in the iris image with visible changes in the average image intensity. Let  $P = \{p_1, \dots, p_k\}, p_i = (x_i, y_i)$ , be a set of primitives and  $d()$  the Euclidean distance. We create an edge from  $p_i$  to  $p_j$  if

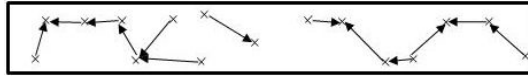
$$e(p_j) < e(p_i) \wedge \frac{e(p_i) - e(p_j)}{d(p_i, p_j)} > \frac{h(p_i, \frac{L}{2}) + h(p_j, \frac{L}{2})}{2} \tag{3}$$

where  $h(p, r)$  is the image entropy of the window centered at pixel  $p$  and with radius  $r$ . As illustrated by figure 2, for each iris image we obtain a correspondent directed graph used to distinguish between identities. It can be observed the high similarity between the graphs resultant of images of the same iris (figures 2(a) and 2(b)). Also, evident differences can be observed in the graph resultant of an image of a different iris (figure 2(c)), either the number of vertices and the edges directions.

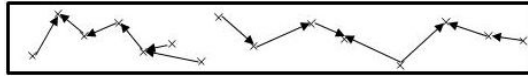
**3.3 Structural Matching**

Having analyzed some of the most cited graph matching proposals, we observed that they commonly enhance the vertices and edges adjacency and do not take into account the relative position of each vertex in the graph, which is very relevant for our purposes. In this compass, Blondel *et al.* [1] proposed an iterative method that measures the similarity between graphs based on the observation that "two graph elements are similar (either vertices or edges) if their neighborhoods are similar".

Our measure of similarity between graphs combines the method of [1] and the minimal distance between the relative position of the vertices (primitives) in each graph. Let  $G_1(P_1, E_1)$  and  $G_2(P_2, E_2)$  represent two directed graphs, with  $P_i$  vertices and  $E_i$  edges. The function  $g()$  gives the dissimilarity between graphs and is used to conclude about the identity of the subjects from where the graphs were constructed.



(a) Graph representing the structure the iris image "1" of subject "1"



(b) Graph representing the structure of the iris image "2" of subject "1"



(c) Graph representing the structure of an iris image of subject "2"

**Fig. 2.** Graphs that represent the structures of the iris images. The upper two figures illustrate the structures of different iris images of one subject and their similarity is evident. Oppositely, the bottom figure represents the structure of an iris image of a different subject and the dissimilarity is notorious.

$$g(G_1, G_2) = \alpha v(G_1, G_2) + (1 - \alpha) e(G_1, G_2) \tag{4}$$

where  $0 \leq \alpha \leq 1$ , can be used to weight the contribution of each term. Through trial-and-error, we empirically considered  $\alpha = 0.6$ . The function  $v()$  gives the minimal distance between the relative position of the primitives in the graph

$$v(G_1, G_2) = \sum_{p_i} \min \frac{d(p_i, p_j)}{|P_1|} + \sum_{p_j} \min \frac{d(p_j, p_i)}{|P_2|} \tag{5}$$

where  $d()$  is the Euclidean distance,  $|P|$  gives the cardinality of the  $P$  set,  $p_i \in P_1$  and  $p_j \in P_2$ .

As described in [1], the computation of the dissimilarity between graphs  $G_1$  and  $G_2$  produces a similarity matrix  $M$  with dimension  $|P_1| \times |P_2|$ , where each cell  $M(i, j)$  gives the dissimilarity between the  $i^{th}$  vertex of  $G_1$  and the  $j^{th}$  vertex of  $G_2$ . The function  $e()$  gives the minimal distance between elements of  $M$ :

$$e(G_1, G_2) = e(M) = \sum_i \min M(i, j) + \sum_j \min M(i, j) \tag{6}$$

where  $i = \{1, \dots, |P_1|\}$  and  $j = \{1, \dots, |P_2|\}$ .

## 4 Experiments and Discussion

### 4.1 Description of Experiments

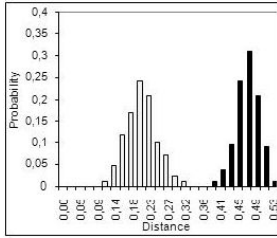
To compare the result obtained by the above described method, we implemented the statistical recognition algorithm proposed by Daugman [3]. This method is usually used as main comparison term for new iris recognition proposals and is the basis of all the commercially deployed iris recognition systems.

It starts by the the iris segmentation, through the integro-differential operator  $\max_{r,x_0,y_0} \left| G_\sigma(r) * \frac{\delta}{\delta r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right|$ . This 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  $(x_0, y_0)$ . Further, to compensate the varying size of the pupil and capturing distances, images are normalized through the previously described *Daugman Rubber Sheet*. The coding of the iris data is accomplished through the use of bidimensional Gabor filters. These spatial filters have the form  $G(x, y) = e^{-\pi\Phi} \cdot e^{-2\pi i\Psi}$ , with  $\Phi = [(x - x_0)^2/\alpha^2 + (y - y_0)^2/\beta^2]$  and  $\Psi = [u_0(x - x_0) + v_0(y - y_0)]$ .  $(x_0, y_0)$  defines the position in the image,  $(\alpha, \beta)$  is the filter width and length and  $(u_0, v_0)$  specify the modulation, having spatial frequency  $w_0 = \sqrt{u_0^2 + v_0^2}$  and direction  $\theta_0 = \arctan(v_0/u_0)$ . To achieve invariance to illumination, the real parts of the 2-D Gabor filters are truncated to zero volume and the sign of the real and imaginary parts from quadrature image projections quantized into binary values. Finally, the Hamming distance is used in the computation of the dissimilarity between iris signatures.

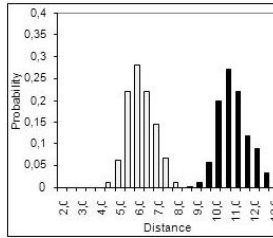
Regarding the choused data sets, there are presently seven public and freely available iris image databases for biometric purposes: CASIA [6], Multimedia University (MMU), University of Bath, UPOL, Iris Challenge Evaluation [8], West Virginia University and UBIRIS. According to the characteristics of each one and to our purposes, we selected two data sets from the most widely used databases for biometric purposes: ICE and CASIA (third version). It should be stressed that the selected images, specially those from the *ICE* database, contain significant noisy regions, ether due to iris obstructions (eyelids or eye-lashes) or reflections (specular or lighting). Each data set contains 800 images from 80 subjects, enabling, respectively, 1800 and 78000 intra- and inter-class comparisons. Further, we divided each one into two halves. The first data sets - *CASIA<sub>tr</sub>* and *ICE<sub>tr</sub>* - were used as training data and the later - *CASIA<sub>tt</sub>* and *ICE<sub>tt</sub>* - to evaluate the recognition accuracy.

### 4.2 Results

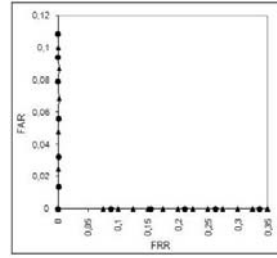
Figure 3 contains a comparison between the results obtained by the statistical Daugman's method and our structural proposal, in the *CASIA<sub>tt</sub>* (upper row) and *ICE<sub>tt</sub>* (bottom row) data sets. The histograms give the probability for the dissimilarity values between the intra- (bright series) and inter-class (dark series) comparisons. Also, the  $\tau$  values correspond to the value of a Fisher-ratio test (FR) given by  $(\mu_E - \mu_I)^2 / (\frac{\sigma_I^2}{t_I} + \frac{\sigma_E^2}{t_E})$ , where  $\mu_I$  and  $\mu_E$  respectively denote the



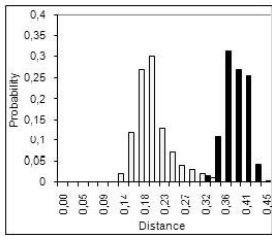
(a) Daugman's method,  $CASIA_{tt}$  data set,  $\tau = 191.39$



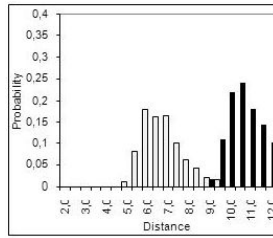
(b) Proposed method,  $CASIA_{tt}$  data set,  $\tau = 161.80$



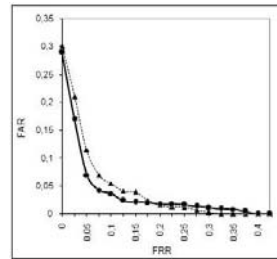
(c) ROCs obtained in the  $CASIA_{tt}$  data set



(d) Daugman's method,  $ICE_{tt}$  data set,  $\tau = 97.11$



(e) Proposed method,  $ICE_{tt}$  data set,  $\tau = 91.58$



(f) ROCs obtained in the  $ICE_{tt}$  data set

**Fig. 3.** Comparison between the results (histograms and ROC's) obtained by the Daugman's algorithm (statistical PR) and our proposal (structural PR) in the  $CASIA_{tt}$  (upper row) and  $ICE_{tt}$  (bottom row) data sets

average of the intra- and inter-class comparisons.  $\sigma_I$  and  $\sigma_E$  are the respective standard deviations and  $t_I$  and  $t_E$  the total of intra- and inter-class comparisons. Finally, the figures of the right column (figures 3(c) and 3(f)) are the obtained receiver operating curves (ROCs) by the Daugman's method (continuous series) and our proposal (dashed series).

Both recognition methods achieved complete separability between the intra- and inter-class comparisons in the  $CASIA_{tt}$  data set and presented error rates equal to 0. Thus, the obtained ROCs fall into the horizontal and vertical axes. Regarding the noisier data set (ICE), the obtained error rates were similar, although the (Fr) values indicate a slightly higher separability between classes by the Daugman's method. However, our proposal achieved a lower value for the false rejections (FRR), when the false acceptances (FAR) were minimized (FRR, FAR=0), which is very relevant, as this is a measure commonly used for the comfort that recognition systems give to the users. These experiments led

us to conclude that, with further optimization, our proposal can constitute an alternative to statistical PR approaches.

## 5 Conclusions and Further Work

In this paper we proposed an iris recognition algorithm which, singularly, follows the structural PR paradigm. Our experiments were performed on two widely used iris image databases (CASIA.v3 and ICE) and show that the proposed iris structure contains enough information to accurately distinguish between individuals. The structural pattern recognition approach reveals itself advantageous essentially to improve the human-perception and description of the PR system's functioning, which can be relevant, for instance, for security / forensic purposes.

Our current work is focused in the evaluation of the robustness that structural-based iris recognition algorithms have to noisy images. We hope to find isomorphisms between sub-graphs, in order to perform biometric recognition based exclusively in the information extracted from small portions of the iris texture, which will substantially increase the operability of iris recognition within noisy image capturing environments.

## References

1. V. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. Dooren (2004) A measure of similarity between graph vertices: applications to synonym extraction and web searching, *SIAM Review* 46, 4:647–666
2. W. W. Boles and B. Boashash (1998) A human identification technique using images of the iris and wavelet transform, *IEEE Transactions on Signal Processing* 46, 4: 1185–1188
3. J. G. Daugman (1993) High confidence visual recognition of persons by a test of statistical independence, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 11: 1148–1161
4. E. R. Dougherty (1992) *An Introduction to Morphological Image Processing*, SPIE Optical Engineering Press
5. Y. Huang, S. Luo, and E. Chen (2002) An efficient iris recognition system, In *Proceedings of the First International Conference on Machine Learning and Cybernetics*:450–454.
6. Institute of Automation, Chinese Academy of Sciences (2004) CASIA iris image database, <http://www.sinobiometrics.com>
7. L. Ma, T. Tan, D. Zhang, and Y. Wang (2004) Local intensity variation analysis for iris recognition, *Pattern recognition*, 37, 6: 1287–1298
8. National Institute of Standards and Technology (2006) Iris challenge evaluation, <http://iris.nist.gov/ICE/>
9. R. P. Wildes (1997) Iris recognition: an emerging biometric technology, *Proceedings of the IEEE*, 85, 9: 1348–1363

---

# Enhancing Real-Time Human Detection Based on Histograms of Oriented Gradients

Marco Pedersoli<sup>1</sup>, Jordi González<sup>2</sup>, Bhaskar Chakraborty<sup>1</sup>,  
and Juan J. Villanueva<sup>1</sup>

<sup>1</sup> Computer Vision Center and Departament d'Informàtica. Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain

marcopede@cvc.uab.es

<sup>2</sup> Institut de Robòtica i Informàtica Industrial(UPC-CSIC), Edifici U Parc Tecnològic de Barcelona. 08028, Spain

**Summary.** In this paper we propose a human detection framework based on an enhanced version of Histogram of Oriented Gradients (HOG) features. These feature descriptors are computed with the help of a precalculated histogram of square-blocks. This novel method outperforms the integral of oriented histograms allowing the calculation of a single feature four times faster. Using Adaboost for HOG feature selection and Support Vector Machine as weak classifier, we build up a real-time human classifier with an excellent detection rate.

## 1 Introduction

Human detection is the task of finding presence and position of human beings in images. Many applications take advantage of it, mainly in the videosurveillance and human-computer interaction domains. Thus, human detection is the first step of the full process of Human Sequence Evaluation [5].

A well known method for detecting and tracking humans in video sequences is the *segmentation of foreground* motion blobs. Modeling static background allows detecting the presence of new objects, as humans, in the scene. This solution is effective when the camera is stationary and background has gradual changes in illumination. However, human detection can be required when using a moving camera (i.e. pedestrian detection in car camera) or in image databases (i.e. human image retrieval). To tackle these problems, an approach based on single image human body detection must be used. This leads to detect humans using *appearance-based* image features. In practice, a feature pattern learnt by a classifier is exhaustively searched in the full image.

Detecting human bodies based on appearance is more difficult than detecting other objects such as cars or faces. Human bodies are non-rigid, and highly articulated. This implies that besides illumination changes, occlusions and perspective distortion, we have to deal with a high range of different poses and

postures. Additionally, in human detection it is not possible to take advantage of specific textures and color information due to the variability of worn cloths.

This paper is organized as follows: section 2 explains previous work on human detection, section 3 gives an overview of the human detection framework that we built and section 4 shows the results we obtained in experimental tests. Finally section 5 discusses the conclusions.

## 2 Previous Work

In literature, two main approaches for human detection based on classifiers have been investigated [3]. The first is the detection of human parts (i.e. head, torso, legs) while the second is detecting humans as a whole. So far, the latter method has shown better detection results. On the other hand, the part-based approach can deal with occlusions and a broader variety of poses.

Regarding body-part based approach, Felzenswalb and Huttenlocher [2] model the human body as a set of human parts joint with springs, where every part is detected using Gaussian Derivative filters. Ioffe and Forsyth [6] represent each body component as a projection of straight cylinders which are then assembled into a full body. Mikolajczyk et al. [8] use orientation-position histograms as representative feature of every body part. Once the parts have been detected, these are assembled with a probabilistic model. Wu and Nevatia [13] introduce edgelets, a new type of silhouette-oriented features for learning body parts within a boosting framework.

In the full body detection approach, Papageorgiou et al. [9] use a feature descriptor based on Haar wavelets and a polynomial Support Vector Machine (SVM) as classifier. Gavrilla and Philomen [4] calculate edge images and compare them with a set of learnt exemplar using Charmfer distance. Viola et al. [12] combine Haar-like wavelets and space-time differences features into an AdaBoost machine to exploit the additional information given by motion. An effective human detector is proposed by Dalal and Triggs [1], who describe humans as a dense grid of Histogram of Oriented Gradient (HOG) and detect them with the help of a SVM. Zhu et al. [14] enhanced Dalal and Triggs results using integral of histograms for a fast HOG computation and Adaboost for feature selection.

In this article, following the same approach of Zhu's, we present a new technique for feature extraction that improves the calculation speed. Then, we use the gained speed to add to the HOG features a gaussian weighting mask which is improving the final detection rate.

## 3 Human Detection Framework

Human detection can be seen as a classification problem. In practice, an image is densely scanned by a detection window that moves to any possible position and with any possible scale, looking for humans. The task is then to construct a good classifier that can discriminate humans and not humans in a really short

time and with very low false positive (because an image can contain from 10000 to 100000 detection windows depending on the density of the search).

Two main subjects are fundamental for human classification: *features extraction* and *classification learning*. The first part consists on extracting the most relevant information from the data available. In our case, we want to find out an optimal image pixel representation that can underline differences between human and not human images. The choice of the feature is done throughout an analysis of the semantic significance of the feature. We choose to use HOG feature because (i) they provide a good representation of silhouettes and borders, (ii) are invariant to light contrast changes and small image movements and (iii) can be computed in a constant time independently of their size.

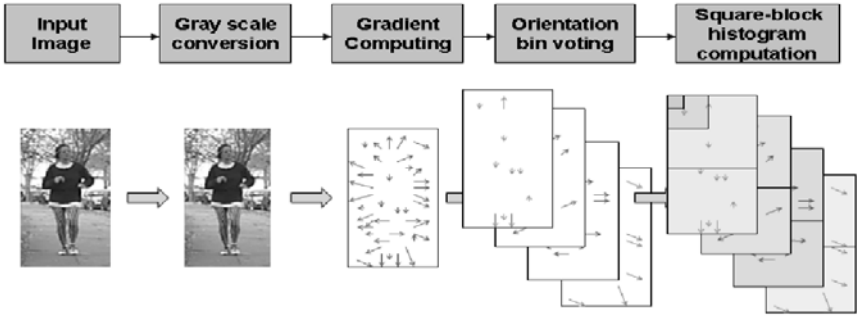
To learn a specific pattern like a human silhouette, it is necessary to associate its corresponding feature pattern throughout a classification process. In our framework, the classification learning is obtained using Adaboost, a boosting technique based on the construction of a strong classifier as a linear combination of weak classifiers. In practice, from human images a set of HOG features with different sizes and positions are extracted and then the best ones are selected for detection. In the next subsections, a detailed explanation of every step of the human detection framework is discussed.

### 3.1 Feature Extraction

Most human detection systems are based on the original idea of Viola and Jones [11]. They, using an overcomplete set of wavelets features (Haar-like features), were the first to get a well performing real-time face detection system. The main contribution of that work is on the ability to build up, throughout AdaBoost, a cascade of classifiers that give at the same time real-time performance and good detection rate. Furthermore, they took advantage of the integral of the image to be able to speed up the Haar-like features calculation. Unfortunately this solution has been proved to be not good enough for human detection. Haar-like features has a too little discriminative power, so that in case of high appearance variability classes, like humans, they can not capture enough information to allow pattern learning. Alternatively recent works [1, 14] showed that HOG (Histogram of Gradients) features can give excellent results in human detection and also that can be calculated in a constant time using the integral histogram [10]. In this paper we propose a novel method for HOG extraction that relies on the recursive calculation of square-block histograms. It can be four times faster than the integral of the gradient.

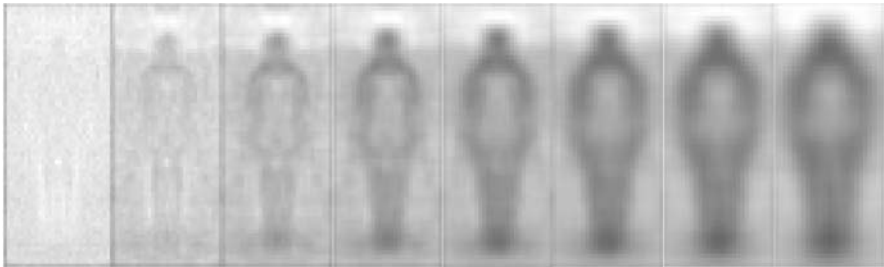
Figure 1 represents the preprocessing necessary for the feature calculation. First the image is converted to gray-scale. Then image gradient module and phase are calculated applying an horizontal and vertical convolution with  $[-1 \ 0 \ 1]$  and  $[-1 \ 0 \ 1]^T$  masks. At the next step, a clustering process based on the gradient module orientation is achieved. In practice the gradient image is split into 8 image bin, each one representing an orientation from  $0^\circ$  to  $180^\circ$ . For every pixel, the corresponding module of the gradient is stored in the appropriate orientation image bin. At this point, we calculate recursively the histogram of square-blocks.





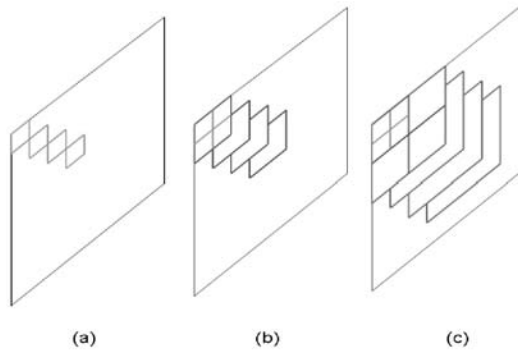
**Fig. 1.** Image Preprocessing: before calculating HOG features is necessary to preprocess the image to obtain square-blocks of histograms which are used for a fast HOG calculation

These blocks are the basic components for the following HOG construction. As it is possible to see in Figure 2, the blocks start to be relevant (in the image it is possible to distinguish a human silhouette) from a size of 4 pixels. Then, this is the basic block we compute. Then recursively, square-blocks of  $2^2, 2^3, 2^4, 2^5$  size are calculated at steps of 4 pixels. As showed in Figure 3, each block can be calculated as sum of 4 blocks of the previous scale level or 16 blocks of 2 previous scale levels. The second case is useful when we want to apply a gaussian mask to the feature. This will be applied in the experiments.



**Fig. 2.** Histograms of square-block variance calculated from 100 images of the MIT human database. From left to right are represented the results using a constant moving step of 2 and different block sizes:  $\{2 \times 2, 4 \times 4, 8 \times 8, 12 \times 12, 16 \times 16, 20 \times 20, 24 \times 24, 32 \times 32\}$ .

After this image preprocessing, the feature calculation can be executed really fast, and in a time that does not depend on the feature size, but constant. In fact, the procedure to calculate a feature consists of recollecting the precalculated histogram blocks belonging to the feature and normalize them. The main difference of this method with the integral of histogram is that each histogram can be obtained with only one memory access instead of four (because the value



**Fig. 3.** Distribution of features: (a)  $4 \times 4$  square-blocks: each block is calculated directly from the gradient image and they are not overlapping (b)  $8 \times 8$  square-blocks: each block is calculated from 4 blocks of the previous level and they are overlapping each other of 50% (c)  $16 \times 16$  square-blocks: each block can be calculated either with 16 blocks of size  $4 \times 4$  or with 4 of size  $8 \times 8$ . They are overlapping each other of 75%.

is directly read from the memory), making the feature extraction process around four times faster.

### 3.2 Classification Learning

Our classification algorithm is based on Adaboost. This is a technique of combining a sequence of weak classifiers (classifier with a classification rate slightly better than 0.5) into a strong classifier. Given a set of samples  $X = \{x_1, x_2, \dots, x_n\}$  and a set of weak classifier  $h_t$  which assigns every element to -1 or +1 whether the element belong to a class or not, Adaboost provides a strong classifier

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

which is a linear combination of  $h_t$ . Boosting technique is obtained iteratively assigning importance weight to the samples used for the training session. At the start all the weight have the same value, but after the first weak classifier has been added, the weight value is updated. The weight of samples with a right classification keeps the same value, while the weight of samples with wrong classification is increased, to give them more importance in the next iteration. In the same way, at every iteration, the classifier coefficient values  $\alpha_t$  are calculated based on the relative goodness of the weak classifier.

In our case, the weak classifiers are Support Vector Machine associated to a single HOG feature, and these are sorted by their own classification capability. When Adaboost is called from the first time for every feature a SVM is learnt. The SVM with the best classification score is selected. At this point Adaboost update the importance weight associated to every sample and calculate the  $\alpha_1$  coefficient. Then, again, for every feature the corresponding SVM is learnt, but this time with different importance weights and then also with different result.

This cycle is repeated until a certain detection rate is reached or until the strong classifier has reached a certain complexity. We considered 50 weak classifiers as a good trade off between detection rate and complexity.

### 4 Experimental Results

We tested our detector with the MIT pedestrian database. This database is composed by 924 full body pedestrian images of size 128x64 pixels. We used 800 of them for the training and the rest for the test. The total training session is then composed by 800 positive images (humans) form the MIT database and 800 negatives (not humans) taken from random windows cropped from no human images. For every sample image we extracted 619 features which are the sum of:

- 29 \* 13 features of size 16x16 pixels
- 25 \* 9 features of size 32x32 pixels
- 17 \* 1 features of size 64x64 pixels

This approach is something like a trade-off between Dalal and Triggs approach which is using a low number (105 per human image), fixed size, computationally slow HOG features and Zhu et al. who are using a high number (5031), variable size and ratio faster HOG implementation.

We did two different kind of tests: the first using normal HOG features exactly as in Zho's, the second applying a gaussian weighting mask to the features. The idea of the second experiment is to use the increased speed obtained form our feature calculation technique to gain better detection performance. In fact, applying to the feature a gaussian mask of standard deviation equal to a half the feature size, reduces the effect of small perspective changes, making it more robust [7]. The gaussian weighted feature can be calculated using 16 square-blocks (instead of 4) of two inferior scale size features and multiply each of them for the corresponding coefficient.

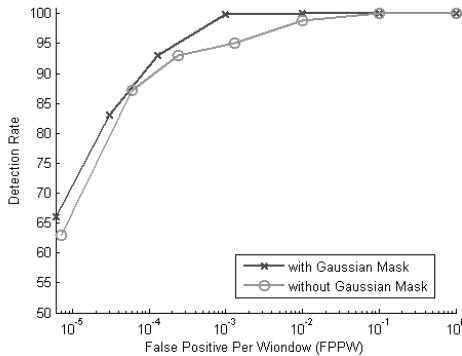


Fig. 4. ROC curve of two detectors with different HOG features

The ROC curves of the two experiments are shown in Figure 4. Both detectors are well performing, with a detection rate at  $10^{-4}$  false positive per window greater than 85%. As expected the features with gaussian weighting are performing better than the others. However, the increment of detection rate in the ROC curve is varying from 1% to 5%, which is probably a too small gain considering the loss of 4X of speed necessary for that.

## 5 Conclusions

In this paper we presented a human detector framework based on a HOG features, which are the actual state of art in the field of detection systems. We defined a new way of computing HOG features based on square-block of histograms which is four time faster than Zhu's implementation based on the integral of histograms. Finally, we exploited this increased speed to apply an approximate gaussian mask to the features in order to improve the feature quality and the corresponding detection rate.

The results described in the previous section show the quality of the approach. Nevertheless, we did not reach the performances of [1] and [14]. We believe that this is due to the reduced number of negative examples we could employ because of the limited amount of memory we could use in our Matlab implementation. Thus, it can be useful to convert at least part of the code in a more efficient language, like C or C++. This allows us to experiment more, and with bigger and more difficult examples. A good testing set could be the INRIA database which has more, and more difficult human position and situation. Finally, Another point to improve the framework is a cascade of classifiers. This can provide a twofold advantage. The speed of the final detector can be highly enhanced using an approach based on subsequent classification refinements. Moreover, in the learning phase of the cascade, difficult examples can be selected to feed the next cascade stage, avoiding the necessity of a high number of negative examples which slows down the learning process.

## Acknowledgements

This work is supported by EC grants IST-027110 for the HERMES project and IST-045547 for the VIDi-video project, and by the Spanish MEC under projects TIN2006-14606 and DPI-2004-5414. Jordi Gonzàlez also acknowledges the support of a Juan de la Cierva Postdoctoral fellowship from the Spanish MEC.

## References

1. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

2. P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, pages 66–73, 2000.
3. D. Gavrilu. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999.
4. D. Gavrilu and V. Philomin. Real-time object detection for “smart“ vehicles. In *CVPR*, pages 87–93, Fort Collins, Colorado, USA, 1999.
5. J. Gonzàlez. *Human Sequence Evaluation: The Key-frame Approach*. PhD thesis, Universitat Autònoma de Barcelona, Spain, 2004.
6. S. Ioffe and D. A. Forsyth. Probabilistic methods for finding people. *Int. J. Comput. Vision*, 43(1):45–68, 2001.
7. D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.
8. K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision*, volume I, pages 69–81, 2004.
9. C. Papageorgiou and T. Poggio. A trainable system for object detection. *Int. J. Comput. Vision*, 38(1):15–33, 2000.
10. F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *CVPR*, 2005.
11. P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision - to appear*, 2002.
12. P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *Int. J. Comput. Vision*, 63(2):153–161, 2005.
13. B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 90–97, Washington, DC, USA, 2005. IEEE Computer Society.
14. Q. Zhu, M. C. Yeh, K. T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1491–1498, Washington, DC, USA, 2006. IEEE Computer Society.

---

# Decisive Factors in the Annotation of Emotions for Spoken Dialogue Systems

Zoraida Callejas and Ramón López-Cózar

Dept. Languages and Computer Systems, University of Granada  
18071 Granada Spain  
{zoraida,rlopezc}@ugr.es

**Summary.** The recognition of human emotions is a very important task towards implementing more natural computer interfaces. A good annotation of the emotional corpora employed by researchers is fundamental to optimize the performance of the emotion recognizers developed. In this paper we discuss several aspects to be considered in order to obtain as much information as possible from this kind of corpora, and propose a novel method to include them automatically during the annotation procedure. The experimental results show that considering information about the user-system interaction context, as well as the neutral speaking style of users, yields a more fine-grained human annotation and can improve machine-learned annotation accuracy by 24.52%, in comparison with the classical annotation based on acoustic features.

## 1 Introduction

Accurate annotation is a first step towards optimized detection and management of emotions, which is a very important task in order to avoid significant problems in communication, as for example misunderstandings and user dissatisfaction, which end up in very low task completion rates. Some studies, e.g. [5], have shown that once the user is in this type of states, it is difficult to guide him out. Furthermore, these bad experiences can also discourage users from employing the system again.

In despite of its benefits, annotation of emotions in spoken dialogue systems has restrictions that issue some important problems to be faced. For example, all information must be gathered through the oral modality and in some systems where the dialogue is less flexible, the length of the user prompts can be too small to use other knowledge sources like linguistic information. As our aim is to use context information even with restricted interactions (e.g. with systems that use system-directed initiatives for dialogue management), we suggest the inclusion of two new different context sources: neutral speaking style of users and dialogue history. The former, provides information about how users talk when they are not conveying any emotion, which can lead to a better recognition of the user non-neutral emotional states. The latter, involves using information about the current dialogue state in terms of dialogue length and number of confirmations and repetitions, which gives a reliable clue about which is the emotional state of the user in each moment.

We have applied this contextual information to the annotation of three negative user emotions: *doubtful*, *angry* and *bored*. The first is useful to know how the dialogue context influences the user certainty about what to do next; whereas the second and third must be recognized before the user gets too much frustrated because of system malfunctions. We consider useful to distinguish between the three because they would involve different dialogue management strategies once the recognizer is implemented.

The rest of the paper is structured as follows. In Section 2, we measure the impact of the proposed contextual information sources over human annotation. In Section 3, we evaluate the performance of our approach with machine learning approaches and compare the results to the ones obtained by the human annotators. To automatically classify emotions, we introduce a novel method in two steps which enhances negative emotion annotation with automatically generated context information. The first step introduces dialogue context and allows the distinction between *angryORbored* and *doubtful* categories; whereas the second calculates users' neutral speaking style, which we use to classify emotions into *angry* or *bored*. Finally, in Section 4 we discuss the conclusions extracted from the experimental results and point out some future work guidelines.

## 2 Human Annotation Results

To obtain rigorous annotations the most reliable way is to recruit specialized annotators, for example psychologists who are trained to recognize human emotions. Unfortunately, in most cases expert annotators are difficult to find and thus the annotation must be done by non-expert annotators [6]. We employed nine non-expert annotators, which is much more than what is typically reported in previous studies [2] [3]. The segment considered for the assignment of emotions was the whole utterance because it was not useful to employ smaller segmentation units (i.e. words) in our case, given that our goal was to analyze the emotion as a whole response to a system prompt, and track its effect on the subsequent interaction, and not studying the change in the emotion within a user utterance.

The utterances corpus employed in our experiments was collected from real users interacting with the UAH (Universidad Al Habla - University On the Line) dialogue system [1]. It is comprised of 85 dialogues, which contain 422 user turns, with an average of 5 user turns per dialogue. The corpus has a similar size to other real emotional speech corpora like those used by [2] (10 dialogues, 453 turns) or [4] (391 user turns). The corpus was annotated twice by every annotator, firstly in an ordered style and secondly in an unordered style. In the first mode the annotators had information about the dialogue context and the system's user speaking style and in the second they did not, so in the unordered style their annotations were based only on acoustic information. The annotation result in both ordered and unordered schemes, was the emotion annotated by more than 4 annotators. If the result was not the same for the ordered and the unordered annotation, then a non-neutral emotion was preferred as global annotation result. In the case in which the results were both non-neutral but

still different, the utterance was discarded. Global annotation results were the tags used for the machine learning approaches.

Emotional corpora extracted from real users interacting with spoken dialogue systems, are usually very unbalanced [2]. In our experiments, the 87.28% of the utterances in the UAH corpus were annotated as *neutral* in the ordered case, whereas in the case of unordered annotation, the corpus was even more unbalanced: 90.68% of utterances were annotated as *neutral*. As shown in Figure 1, the ordered annotation style yielded a greater percentage for the *bored* category, concretely 39% more than in the unordered style. In addition, the *angry* category was substantially affected by the annotation style (i.e. ordered vs. unordered), concretely 70.58% more *angry* annotations were found in the ordered annotation style. On the contrary, the *doubtful* category was practically independent from the annotation style: only 2.75% more doubts were found in the unordered annotation.

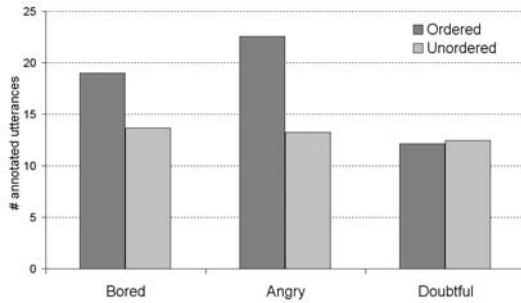


Fig. 1. Proportion of non-neutral annotated utterances

The reason for these results is that taking into account context in the ordered case causes a more stable annotation per dialogue; for example if anger is detected in one prompt then the next one is probably also annotated as *angry*. Besides, the context allowed the annotators to have information about the user speaking style and the interaction history. On the contrary, in the unordered case, they only had information about the current prompt. Hence, sometimes they could not distinguish whether the user was angry or he normally spoke loud and fast. Thus, it is an important fact to be taken into account when annotation is carried out by non-expert annotators. Furthermore, when listening to the corpus in order, the annotators had information about the position of the current user turn inside the whole dialogue, which also gave a reliable clue about the user state.

### 3 Machine-Based Annotation Results

Our first set of experiments was carried out to try to classify emotional prompts (i.e. not tagged as *neutral*), considering acoustic information only. This is a



classification problem that can be solved by different machine learning algorithms that receive as input tuples of features related to acoustic information. We decided to use 60 features after a literature survey to find the most employed by authors [3] [4]. These are utterance-level statistics corresponding to the four groups set out in Table 1.

**Table 1.** Acoustic features used for classification

Category	Features
Fundamental frequency (F0)	Min, max, range, mean, median, standard deviation, slope, correlation coef., regression error, value at first voiced segment, value at last voiced segment
F1, F2, B1, B2	Min, max, range, mean, median value at first voiced segment, value at last voiced segment
Energy	Min, max, range, mean, median, standard deviation, slope, correlation coef., regression error, value at first voiced segment, value at last voiced segment
Rhythm	Rate, voiced duration, unvoiced duration, value at first voiced, number of unvoiced segments

The first group was comprised of pitch features, which are significant indicators for emotional speech when compared to neutral conditions. We calculated all the pitch features in the voiced portion of speech. All the duration parameters (e.g. slope) were normalized by the utterance duration to obtain comparable results for all the utterances in the corpus. The second group was comprised of features related to the first two formant frequencies (F1 and F2) and their bandwidths (B1 and B2). Different speaking styles produce variations of the typical positions of formants. In the particular case of emotional speech, the vocal tract is modified by the emotional state. Energy was considered in the third group of features, it is related to the arousal level of emotions. The variation of energy of words or utterances can be used as a significant indicator for various speech styles, as the vocal efforts and ratio (duration) of voiced/unvoiced parts of speech changes. For these features, we only used non-zero values of energy, similarly as what we did for pitch. Finally, the fourth group was composed by rhythm features. Rhythm and duration features can be good emotion indicators as the duration variance decreases for most domains under fast stress conditions.

In our case the most frequent emotion category was *angry*, so the first machine learning approach that we used for comparison purposes was a baseline that always annotated user turns with this label. Secondly, we used the feature vectors as an input to a multilayer perceptron (MLP) classifier, which we used for our experiments following a 10-fold cross-validation strategy.

### 3.1 Classification Based on Audio Features

When we used the traditional classification based on audio features, the emotion recognition rate was 51.62% for the baseline, whereas for the perceptron

was 35.48%. These results are comparable to the case where human annotators labeled user turns unordered.

It was possible that not all the features employed for classification (60 in total) were very informative. As using irrelevant features makes the learning process slower and increases the dimensionality of the problem, we carried out a feature selection. Three methods were employed for feature selection: a forward selection algorithm, a genetic search, and finally a ranking of the features (instead of finding a subset) using the information gain as a ranking filter. The optimal subset, as it appeared with non zero gain in all the three approaches, seemed to be comprised of B1 in the last voiced segment, and energy maximum. However, we obtained no improvements using only the selected features as the accuracy was again 35.48% for the multilayer perceptron.

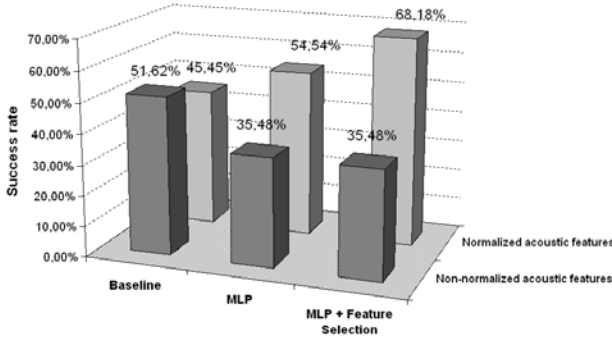
### 3.2 Classification Based on Normalized Audio Features

When humans annotated the corpus in ordered style they had information about previous user turns. Therefore, they could know e.g. that user 'A' always speaks very fast and loudly, whereas user 'B' always speaks relaxed. Therefore, some acoustic features may be the same for 'A' neutral as for 'B' angry. Hence, if the algorithms automatically learn that these features values correspond to the *angry* category, the classification will fail for user 'A'.

To solve this problem we had the user context into account, and normalized the features around the neutral voice of the user. To do this, we calculated the user's neutral voice features in each dialogue and subtracted those from the feature's values obtained in the rest of the utterances. In the experiments we considered that the first utterance of the user was neutral, assuming that he was initially in a non-negative emotional state. This assumption is feasible as employing other approaches like average value of utterances is impossible to calculate in an emotion recognizer working on real time. Thus, the first utterance of each dialogue was not taken into account for the classification experiments, that is why the baseline accuracy obtained is different across them, even when we used the same dataset. In this case we obtained 45.45% correctly classified utterances for the baseline and 54.54% with MLP.

Using the features selected in the previous section (B1 in the last voiced segment and energy maximum) we obtained 68.18% correctly classified utterances (13.64% more compared to no feature selection). In the non-normalized case the feature selection did not introduce any improvement. Thus, using normalized acoustic features (68.18% success rate) yielded an improvement of 16.56% compared to the best case in non-normalized classification (51.62% success rate), which is achieved with the baseline algorithm (Figure 2).

A study of the confusion matrices of all the described experiments showed that the *doubtful* category is often confused with the *angry* or *bored* categories, with percentages above 20% in most cases. Thus, automatically learned annotations are affected by the context information in the same way than human annotations, as for them ordered annotation did not improve the annotation of the *doubtful*



**Fig. 2.** Percentage of correctly annotated turns with normalized vs. non-normalized acoustic features

emotion. In contrast, for human annotators adding context information (ordered case) lead to better results in the annotation of *angry* and *bored*.

To confirm that better results can also be obtained with automatic approaches by deciding only between *angry* and *bored* categories, we classified only *bored* and *angry* utterances with the multilayer perceptron and obtained 71.42% of correctly classified utterances. To improve these results we carried out a new feature selection using the forward algorithm and obtained a subset comprised of three features: F0 median, energy maximum and duration of the longest voiced segment. The classification accuracy was 85.71%.

With these experiments we have shown that normalized acoustic features are preferable to non-normalized, as these yield 16.56% improvement (68.18% vs. 51.62% success rate). This is due to the information about the neutral style of the speaker. Besides, these results can be improved by 17.53% if we only distinguish between *bored* and *angry* emotions (85.71% with *bored* and *angry* after feature selection vs. 68.18% with three emotions). Thus, we have 85.71% correctly annotated utterances in the best case (MLP classification of *bored* and *angry* with information about the acoustic neutral after feature selection) and 35.48% in the worst case (MLP classification of non-normalized acoustic features regardless of feature selection).

### 3.3 Dialogue Context Annotation

We carried out dialogue context annotation considering two labels: *depth* and *width*. The former indicates the length of the dialogue, whereas the second denotes the number of user turns necessary to obtain a particular piece of information. To obtain the information about the dialogue context we employed the dialogue history, using the system prompt to automatically calculate the value for *depth* (D) and *width* (W).

The following annotation scheme was employed: D was initialized to 1 (0 would mean that the user hangs up the telephone before he says anything) and

incremented by 1 for each new user turn and each time the interaction went backwards (e.g. to the main menu).  $W$  was initialized to 0 and incremented by 1 for each user turn generated to confirm, repeat data, disambiguate input or ask the system for help. For classification purposes we used an *accumulated width* ( $A$ ), so that in dialogue turn  $i$ ,  $A(i)$  was the summatory of the  $W$  values from the first utterance to  $i$ . This way, confirmation and repetition subdialogues in which the user had been involved through the dialogue had always a negative impact on the user emotional state, even if he was not currently in these subdialogues.

An exhaustive study of our corpus showed that in the corpus the distribution of the *angry* and *bored* emotions regarding *depth* and *width* was rather random, e.g. we find users angry or bored with a high *depth* value. Because of this reason, we decided to take into account only two emotion categories: *doubtful* and *angryORBored*, and for classification we implemented an algorithm based on a threshold. The classification algorithm was calculated using the equation:  $T = D + A$ , where  $D$  denotes *depth* and  $A$  the *accumulated width*. A value of  $T$  greater or equal than the threshold indicated *angry* or *bored* emotional states, whereas a smaller indicated *doubtful*. Several values for the threshold were studied, obtaining that  $T = 4$  was the optimal, for which 70% utterances were correctly annotated.

## 4 Conclusions and Future Work

We have carried out several experiments to study the annotation of human emotions in a corpus collected from real (non-acted) interactions with a dialogue system. This is a very difficult task, as even human beings may consider different emotions for a particular utterance. Because of this, many previous studies have focused on the recognition of emotions expressed by actors, who tend to emphasize them.

The experiments consider both a manual annotation from nine human annotators, and automatic annotation with different machine learning approaches. The results show that traditional annotation methods, based solely on acoustic features, yield to worse results in terms of classification error (in the case of automatic annotation) and decrease by 3.4% the number of non-neutral emotions annotated (in the case of human annotation). For machine-learned classification methods, the experimental results show that similarly as what happened with human annotators, the emotion annotation is substantially improved when adding information about the user neutral voice and the dialogue history. The dialogue context is useful to distinguish between *doubtful* and *angryORBored* categories with a 70% success rate. Once an utterance is classified as *angryORBored*, the normalized acoustic features let us distinguish between *bored* and *angry* with 85.71% success. Thus, 60% classification rate can be attained for the three emotions (*angry*, *bored* and *doubtful*), which is 24.52% better than the case in which no context information was used for the annotation.

As our classification method is automatic and can be employed during the running of a dialogue system, main future work guideline will be the design of

an emotion recognizer following this scheme for the UAH system, as an attempt to better adapt automatically its behavior considering the recognized emotional state of the user.

## References

- [1] Callejas, Zoraida, & Ramn Lpez-Czar 2005. Implementing modular dialogue systems: a case study. In *Proceedings of the ASIDE 2005*.
- [2] Forbes-Riley, Kate, & Diane J. Litman 2004. Predicting emotion in spoken dialogue from multiple knowledge sources. In *Proceedings of the HLT-NAACL 2004*, pages 201–208.
- [3] Lee, Chul Min, & Shrikanth S. Narayanan 2005. Toward detecting emotions in spoken dialogs. *IEEE transactions on speech and audio processing*, 13(2):293–303.
- [4] Morrison, Donn, Ruili Wang, & Liyanage C. De Silva 2006. Ensemble methods for spoken emotion recognition in call-centers. *Speech communication*. In Press.
- [5] Riccardi, Giuseppe, & Dilek Hakkani-Tr 2005. Grounding Emotions in Human-Machine Conversational Systems. *Lecture Notes in Computer Science*, pages 144–154.
- [6] Vidrascu, Laurence, & Laurence Devillers 2005. Real-life emotion representation and detection in call centers data. *Lecture Notes on Computer Science*, 3784:739–746.

---

# A Hybrid Big Rock Detection Algorithm Based on Multiple Images Fusion and Watershed

Mohammad T. Al Modarresi, Mohammad S. Tabatabaei,  
and Mohammad T. Sadeghi

Signal Processing Research Lab., Department of Electronics, University of Yazd,  
Iran

[almodarresi@zepler.org](mailto:almodarresi@zepler.org), [m.sadeghi@yazduni.ac.ir](mailto:m.sadeghi@yazduni.ac.ir)

**Summary.** Big rocks are an important problem in the mining industry. They could block machines and cause high costs of preservation equipment. This problem reduces the tonnage productivity and increases the machine down-time. This paper utilizes marker based watershed algorithm for detecting big rocks. First, an inverted binary of the original picture is created. Then the regional maximum distance function of this image is used to mark the rocks in the image. Finally, image fusion on decision level is used to improve rate of big rock detection. Proposed algorithm is examined with a data set that involves 40 big rocks and calculated rate of big rock detection. Result indicated multiple images fusion could decrease error in big rock detection.

## 1 Introduction

Big rock detection is a problem in the mining industry. Identify big rock is increasingly becoming important issues in the materials mining, comminution, and in construction industries utilizing stone products because big rocks can block crushers, vibrator feeders or conveyor belts. Such situations lead to high costs of preservation equipment, reduction of tonnage productivity, increase down-time and may be dangerous for human operators [1, 2, 3].

Visual inspection and sieving are traditional approaches that apply to identify big rock. Image processing techniques provide a quick, non-contact and inexpensive method to determine big rock. Information of the rock surface images can be used to detect the big rocks. Problem of big rock detection can be divided to three stages:

- perform a segmentation of the surface rock to determine the individual regions
- calculation surface of regions
- Identification of big rocks.

Different algorithms apply to identify big rock. Maerz Has applied image classification algorithms to calculate size distribution and shape of rock fragments [2]. Cabello mixed image processing techniques and neural network to identify big rock [3]. In this paper, big rocks are detected using a hybrid of watershed algorithm and multiple image fusion.

In the next section watershed segmentation algorithm is introduced. Then implementation of watershed is described in section 3. Section 4 deals with image fusion that applied on decision level to improve rate of big rock detection. Experiment results described in section 5. Conclusions are drawn in the final section.

## 2 Watershed Segmentation

Image segmentation is one of the pre-processing steps of image analysis. Main goal of segmentation is extracting objects from image. Proposed algorithms can categorized to region-based and histogram-based methods [4].

Overlap and touch of rocks are two important features in the images. Surface of rocks which produce in blasting process, involve various edges and don't have particular shape. Also texture and colour of rocks are similar (fig. 1). So algorithms such as edge detection, thresholding are not suitable for segmentation of these kinds of images. On the other hand, shadow is a parameter that can useful in segmentation but it is a source of error too. Watershed segmentation

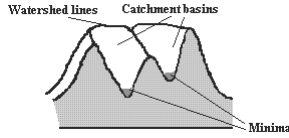


**Fig. 1.** Original image

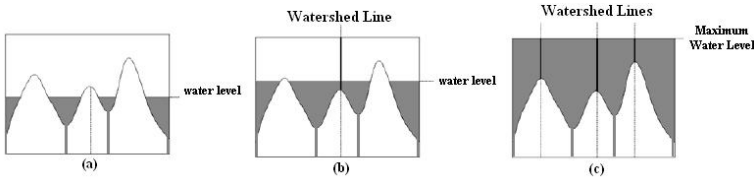
is a famous algorithm and is very useful and powerful tools in many different application fields [4]. The concept of watershed is well known in topography and is based on visualizing an image in three dimensions. Third component of third dimension space is gray level of pixel. In this space, points classified to three types:

1. Points belong to a regional minimum.
2. Points which if a drop of water placed on them, the drop fall down to a single minimum. Set of these points is termed the catchment basins.
3. Points which if a drop of water placed on them, the drop can fall down to several minimum. Set of these points are called divide line or watersheds or crest line (fig. 2).

The classical idea for building the watershed is illustrated Figure 3. Process begin by punched each regional minima of the surface. Then immerse the surface into a lake at a uniform rate. The water progressively floods the basins



**Fig. 2.** Regional minima, catchment basins and watershed lines



**Fig. 3.** Building watershed

corresponding to the various minima (fig. 3.a). To prevent the merging of two different waters originating from two different minima, a dam between both lines is built (fig. 3.b). Once the surface is totally immersed, the set of the dams are the watershed of the image. In one dimension, the location of the watershed is straightforward: it corresponds to the regional maxima of the function (Figure 3.c). In two dimensions, dam boundaries correspond to the ridge of high lands dividing areas. So, boundaries extracted by watershed segmentation algorithm are continuous.

Digabel and Lantuéjoul [5] introduced watershed transformation for segmentation of binary image in 1983. Then Lantuéjoul and Bucher apply it on gray level images. Many algorithms that proposed to calculate the watershed transformation were slow or inaccurate. So, this transformation didn't apply in practical application. At 1991, Vincent and Soile introduced another algorithm based on immersion process. It was very fast and flexible approach [6]. Performance of Vincent's algorithm was good therefore effort to improve algorithms stopped and researchers look for application of algorithm in industries, medicine, and other fields.

The existence of four regional minimum in Figure 3 cause the surface is divided to four regions using watershed algorithm. In general, watershed algorithm created independent region for each minimum. Note that presence noise and undesirable minimums lead to over-segmentation of the image.

Several methods have been proposed to overcome the over-segmentation such as region-growing techniques and Marker-based method [6]. The later method is based on the concept of markers. Marker is a connected component belonging to an image that characterized with features of object (e.g. size, shape, location, brightness, color  $\check{E}$ ) in image. In other words, marker image is binary image which there are connected components for each object that must be segmented.



Maximum (minimum) regional is a feature that many literatures used to created marker image but this marker is suitable for images that objects surrounded with thin dark (light) region. Since shadow is critical feature in rock images, maximum (minimum) is not a suitable criterion to mark this type of images.

As an alternative distance function is used to mark objects this type of images. In a binary image, the value of distance function for each pixel is defined as the Euclidean distance of the pixel to closest light pixel. Therefore, value of distance function of light pixels is zero. The distance function maps binary images to gray-level images [7]. Regional maximum distance function of binary image can be applied to mark the objects in the image. Touching and overlapping objects can separate with this function. So marker extracting procedure needs the creation of binary image from gray-level image. Next subsection describes method of binary image creation.

## 2.1 Binary Image

One important stage in marker extraction procedure is the creation of a binary image from a gray-level image. Several methods such as threshold technique, k-means clustering and Gaussian mixture model are proposed to produce binary image from original image [9]. Threshold technique gives better result than other methods in this research. Therefore, threshold technique is choose to create binary image. In binary image, regions which belonging to stone cluster labeled with one and non-stone cluster labeled with zero value.

### Threshold Technique

Threshold technique is a famous technique in image segmentation algorithms because of intuitive properties and simplicity of implementation. Suppose  $f(x,y)$  correspond to gray-level of pixel  $(x,y)$  in gray-level image and  $g(x,y)$  correspond to value of pixel  $(x,y)$  in thresholded image (binary image).  $g(x,y)$  is defined as:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{if } f(x, y) < T \end{cases} \quad (1)$$

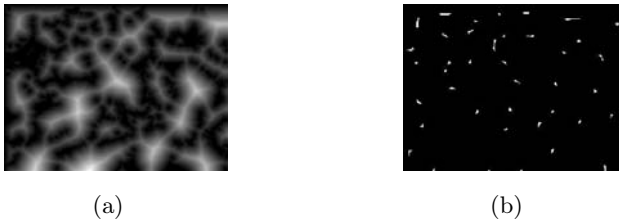
Threshold level (T) is essential parameter in threshold technique. In this research, value of T determined empirically based on data set of image.



**Fig. 4.** Binary image of figure 1

### 3 Implementation of Watershed Segmentation

Watershed algorithm is a powerful tool for image segmentation but over-segmentation is a critical problem in this algorithm. Marker based watershed segmentation is a solution to overcome to this problem. In this algorithm, each rock must be uniquely marked. Regional maximum distance function of inverted binary image is suitable feature to marked stones in image (Fig. 5).



**Fig. 5.** (a) Binary image created with threshold method. (b) Distance function of the inverted binary image.

Then marker image superimposed on base image. Superimpose of marker image on base image implemented with geodesic reconstruction that proposed by Vincent . The regional minima of final modified image are exactly located on the connected components of marker image and others minima of base image were removed. The watersheds of final modified image provide the desired segmentation.

In many literatures, base image was original image or gradient of original image. Distance function of binary image is used for base image at this research.

Watershed algorithm is applied on final modified image. Big rocks are detected by calculation of the area of segmented region. In this investigation, rocks identified as big rocks if their area are more than 4500 pixels. The results of segmentation and Detected big Rocks are shown in Figure 6. In next section, image fusion and fusion on decision level are described. Then a new algorithm is proposed which uses these techniques on the result of segmentation in order to improve big rock detection rate.

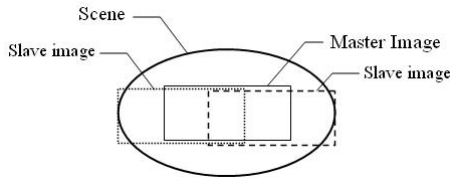


**Fig. 6.** (a) Segmentation result. (b) Detected big rocks.

## 4 Image Fusion

Data fusion techniques combine data from different sources together. The main objective of employing fusion is to produce a fused result that provides the best detailed and reliable information. Also fusing multiple information sources produces a more efficient representation of the data. Data fusion techniques can be categorized into three levels: signal level, feature level, and decision level. Decision level fusion combines the results from multiple algorithms to yield a final fused decision. Image Fusion produces a single image by combining information from a set of source images together. The fused image contains greater information content for the scene than any one of the individual image sources alone. Image fusion requires that images are registered before they are fused [10].

Three images are provided by 3 cameras from a scene on different viewpoints: master image and two slave images. Slave images cover at least 60% of master image (figure 7). Each slave image contains complementary information to master image. So the reliability of the master image increases and accuracy of identified big rocks is improved.



**Fig. 7.** Scheme of master and slave images

The new algorithm works as follows: three images are segmented by watershed method and big rocks candidate are determined in all three images. Then result of segmentation is fused on decision level. Big rocks Candidate in an image are identified as a big rock if they are found in two of images. Flowchart of proposed algorithm is shown in fig. 8.

## 5 Experimental Result

The inspection of segmented image reveals that an object that is identified as a big rock has one of the following states:

- a. It is a big rock and it is identified as a big rock in segmentation procedure (Correct Detection).
- b. It is a big rock but it is divided into small rocks in segmentation procedure (No Detection).
- c. It is a collection of small rocks but it is identified as a big rock in segmentation procedure (False Detection).

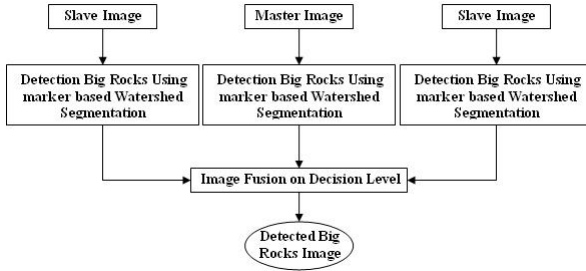


Fig. 8. Flowchart of proposed algorithm

States a and c are recognized as segmentation error.

The main aim of this research is big rock detection. Therefore, rate of correct big rock detection is a suitable criterion for evaluation of proposed algorithms. Also rate of correct detection and rate of incorrect (false) detection must be measured for evaluation of procedure.

The proposed algorithm where examined by a data set that contains 60 gray-level images and Ground Truth of them. Images of data set contain 40 big rocks that characterized with ground truth. Ground Truth is a binary image that created manually (fig. 9). Rate of correct detection and false detection for two methods are shown in table 1.



Fig. 9. Ground truth of big rocks in figure 1

Table 1. Rate of correct and false detection

method	correct detection( )	false detection
big rock detection with single image	85	24
proposed algorithm	92.5	8

Results indicate that image fusion on decision level can improved rate of correct detection. Slave images involve extra information. So image fusion decreased rate of false detection.

In future work, boundary of detecting big rocks in proposed algorithm will be used as initial contour for an active contour (snake) algorithm to reduce rate of false detection [11].

## 6 Conclusion

Big rock detection is a challenging problem in mining industry. A novel algorithm is proposed which fused information of three images on decision level. Images are registered before they are fused. Results indicate image fusion on decision level can increase the rate of correct detection and decreases the false detection rate.

## References

1. Cabello E, Sánchez M A, Delgado J (2002) Real-Time Imaging 8:1–9
2. Maerz N H (1996) Proceeding of fifth FRAGB AST, 115-120
3. Thurley M T, Ng K C (2005) Computer Vision and Image Understanding 98:239–270
4. Gonzalez R C, Woods R E (2001) Digital image processing. Prentice-Hall, New Jersey
5. Digable H, antue oul C (1987)Proceeding in Euro. Symp. Quantitative analysis of microstructures in material science, biology and medicine 85-99
6. Vincent , Soille P (1991) IEEE Trans. Pattern analysis and machine intelligence 13(6):583–598
7. Paglieroni D (1992) Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing 54(1):57–58
8. Vincent (1993) IEEE Trans. on Image Processing 2(2):176-201.
9. Jain A K, Murty M N, Flynn P J (1999) ACM Compu. Surv. 31(3):264-323
10. Wansas N (2003) Feature based architecture for decision fusion. PhD Thesis, Waterloo University, Canada
11. Kass M, Witkin A, Terzopoulos D (1988) computer vision 321–331

---

# Design Description Hypergraph Language

Ewa Grabska, Grażyna Ślusarczyk, and Michał Glogaza

The Faculty of Physics, Astronomy and Applied Computer Science  
Jagiellonian University, Reymonta 4, 30-059 Kraków, Poland  
uigrabsk@cyf-kr.edu.pl, gslusarc@uj.edu.pl

**Summary.** This paper deals with problems of knowledge-based decision support design systems. A new method of transformation the design knowledge into a computer internal representation is proposed. The presented approach is illustrated on examples of designing floor-layouts, where a layout internal representation is generated step by step during the design process. Such a representation in the form of a hypergraph is a data structure allowing for automatical processing of the design project. The prototype software to translate layouts into hypergraphs written in Java is used.

## 1 Introduction

It is well known that there exist a lack of consistency between the given domain knowledge (architecture, construction, machine building) and its internal representation in a computer programme. In order to fill this gap the intermediate tools should be built [2]. On the input they should take data about the designed object in the given domain oriented language and give on the output data structures suitable for automatic processing. This paper can be seen as an attempt to solve the problem of transformation the design knowledge into a computer internal representation for computer-aided design systems. The proposed approach is illustrated on examples of designing floor-layouts, where a layout internal representation is generated in an automatic way.

In this paper we develop a layout language, which supports innovative designing of floor-layouts. This language is called a *diagram language* and its elements are named *design diagrams*. The syntactic design knowledge of the diagram language is defined by means of hierarchical hypergraphs [1], which constitute internal representations of design diagrams. Hypergraphs are a specific type of graphs which allow for expressing multi-argument relations between components of design diagrams.

All modifications of the design diagram made by the user are reflected in the hypergraph representation, which is correspondingly changed using operations defined on hypergraphs. When the designer decides to divide a chosen area of the layout into smaller parts the operation called *hyperedge development* is used. In this paper we also propose a reverse operation on hypergraphs called *hyperedge suppression*. It is useful when the designer wants to redesign the chosen area.

The data structure in the form of a hypergraph can be useful in a knowledge-based decision support system. Based on the syntactic analysis of solutions being created the system can support the designer by suggesting further steps and prevent him/her from creating designs not compatible with specified restrictions or criteria.

## 2 Design Diagrams and Hierarchical Hypergraphs

In our approach two representations of layout are considered. The first one is a simplified architectural drawing called a design diagram. It is composed of rectangles representing components of a floor-layout, like functional areas or rooms. Dashed lines separating rectangles denote the accessibility relations between areas, while continuous lines shared by rectangles denote the adjacency relations

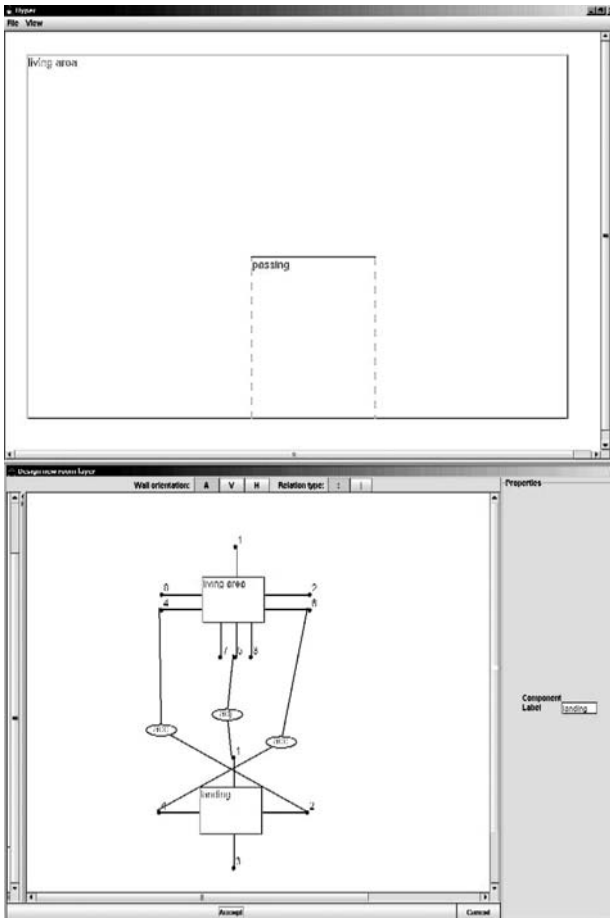
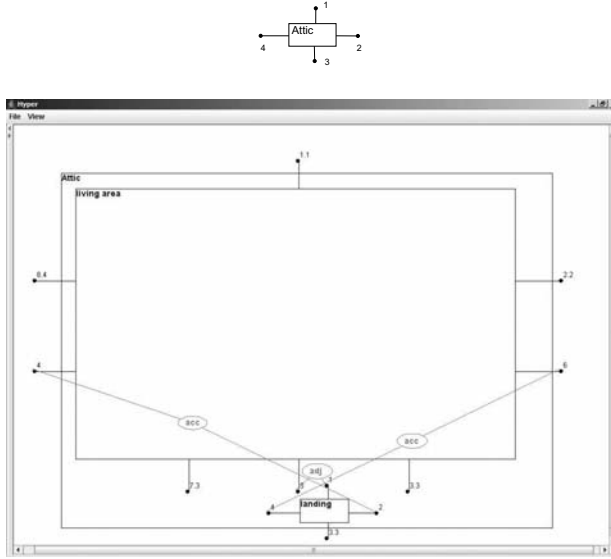


Fig. 1. a) Two parts of the attic, b) the corresponding layout hypergraph



**Fig. 2.** a) The initial hypergraph, b) the hierarchical layout hypergraph obtained by nesting the hypergraph  $h_1$  in the hyperedge of  $h_0$

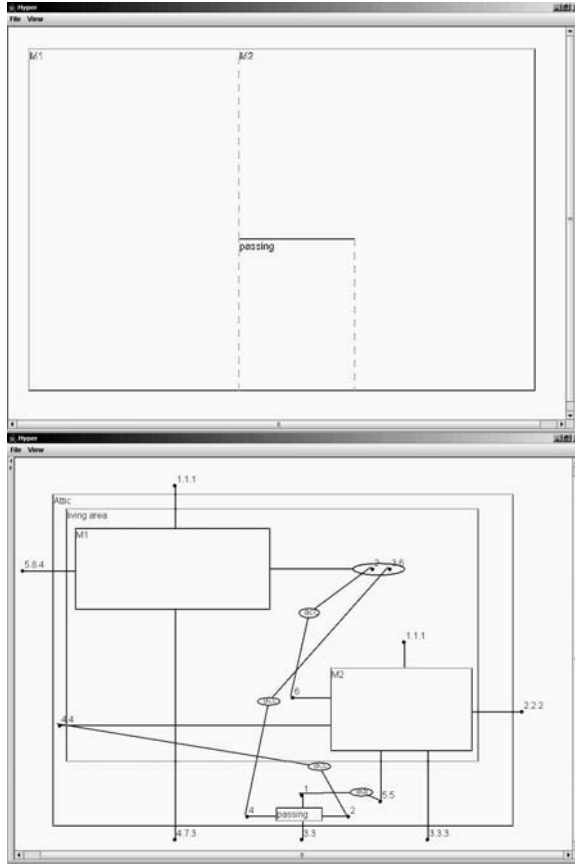
between them (Fig.1a). The second representation of layout, which is a design diagram internal representation, has a form of a hierarchical layout hypergraph (Fig.2b). This hypergraph contains two types of hyperedges, which can represent components and spatial relations on different levels of details. Hyperedges of the first type are non-directed and correspond to layout components like rooms which elements on the lower level of detail can be treated as walls, doors, windows etc. Hyperedges of the second type represent relations among components.

In this paper the proposed representation [1] is extended to contain also hierarchical nodes (Fig.3b). While a hypergraph node corresponds to a single wall, a hierarchical hypergraph node corresponds to a wall which is created during design by connecting consecutive collinear walls, which are represented as hypergraph nodes nested in a hierarchical one. Hyperedges of the layout hypergraph are labelled by names indicating the corresponding components or relations. A sequence of source and target nodes is assigned to each hyperedge and express potential connections to other hyperedges. Moreover, for each hierarchical hypergraph a sequence of external nodes is determined.

### 3 Operations on Hierarchical Layout Hypergraphs

Operations acting on hierarchical layout hypergraphs enable to express changes which take place during the design process. They allow to create and modify hypergraphs representing structures of objects being designed.

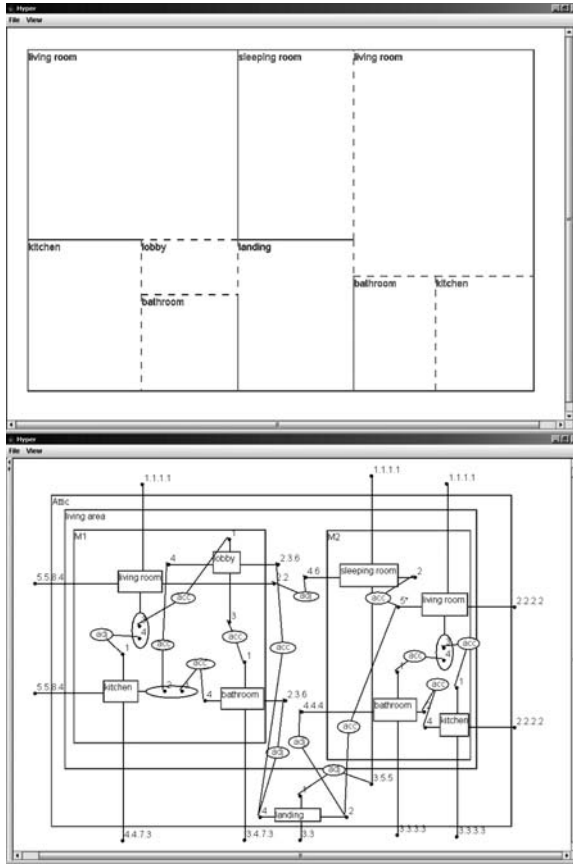




**Fig. 3.** a) The attic with the living area divided into two flats layout, b) the hypergraph representing the divided area

The *hyperedge development* operation, which enables to represent a model structure on a more detailed level, takes two hierarchical layout hypergraphs  $G$  and  $H$  as arguments. As the result of this operation the hypergraph  $H$  is nested in one of the component hyperedges ( $\tilde{e}$ ) of  $G$ . The nodes connected to a developed hyperedge in  $G$  are substituted by the corresponding external nodes from the child graph  $H$  (Fig.3b). The correspondence among nodes is established on the basis of the design constraints represented by the design diagrams corresponding to hypergraphs  $G$  and  $H$ . The development labelling function is defined in such a way that the reverse operation, namely *suppression*, is possible.

The reverse of a development operation is a *hyperedge suppression*. The possibility of suppressing a hyperedge contents is specially useful in modelling complex objects as it enables to consider some object parts on a higher level of abstraction. A suppression operation takes as an argument a hierarchical



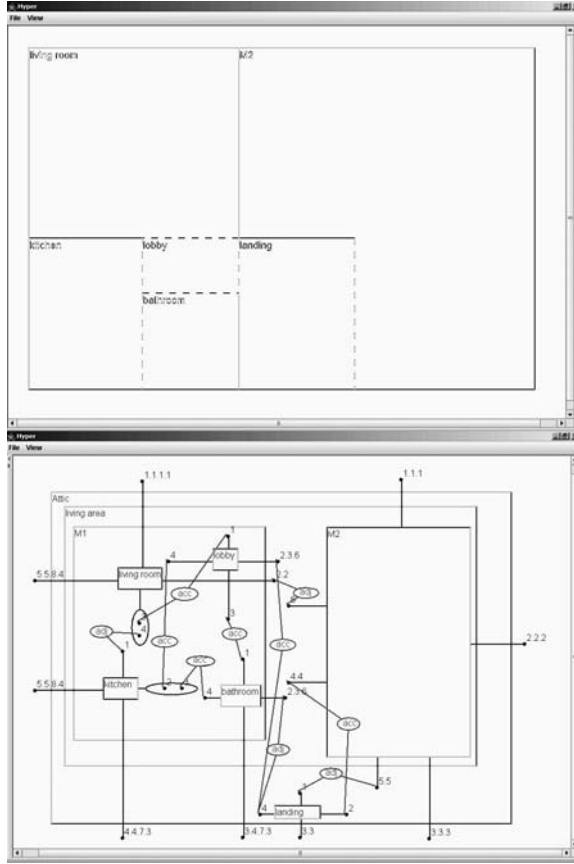
**Fig. 4.** a) The attic with the divided flats, b) the hierarchical layout hypergraph representing the divided area

layout hypergraph and removes the nested contents of one of its earlier developed component hyperedges.

As the result of the hyperedge suppression operation the nested hypergraph  $H$  is removed from its parent component hyperedge  $\tilde{e}$  of a hypergraph  $G$ . The external nodes of  $H$  are substituted by the new corresponding nodes which become new source nodes of the hyperedge  $\tilde{e}$ . The suppression labelling function is defined in such a way that source nodes of  $\tilde{e}$  have the same labels as before development operation on  $\tilde{e}$ . The example of using the development and the suppression operations is presented in the next section.

## 4 Case Study

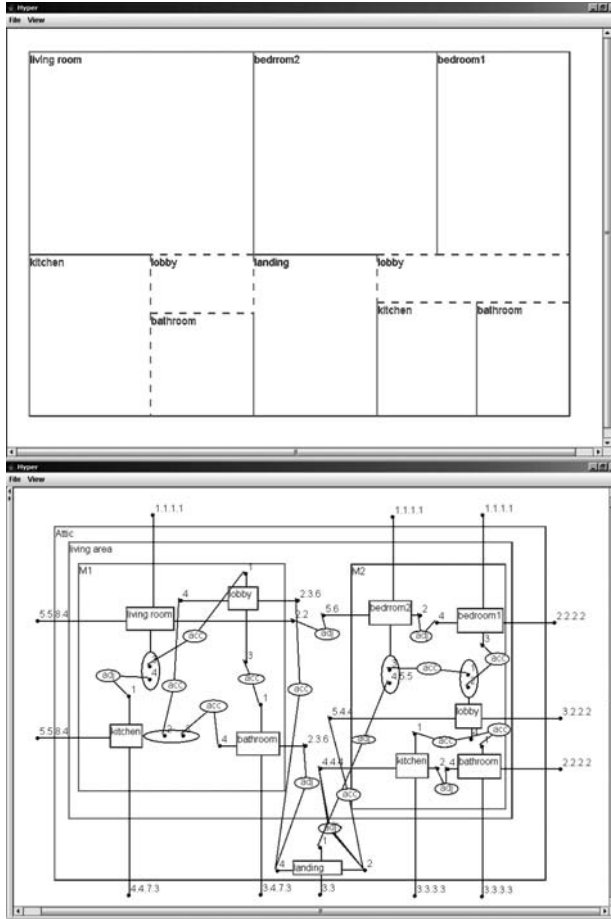
The program called *Hypergraph Generator Supporting Layout Design (HGSLD)* is written in Java 5. The first module of this program enables the designer to



**Fig. 5.** a) The attic with the one flat divided, b) the corresponding hierarchical layout hypergraph

edit the design diagram and reflects the made modifications using operations on diagram’s hypergraph representation. The next module will prompt the designer solutions based on knowledge stored in the hypergraph representation.

Let us consider an example of designing two flats in the rectangular area of an attic. Firstly the designer draws the initial rectangle with the use of the editor module of the program *HGSLD*. At the same time the initial hypergraph  $h_0$  representing this shape is automatically generated (Fig.2a). This hypergraph is composed of one hyperedge connected with four external nodes representing sides of the area. The nodes are numbered in a clock-wise manner. In the next step the designer separates the living area from the floor landing (Fig.1a). The program generates the new hypergraph  $h_1$  (Fig.1b) corresponding to the modified area. We can have two relations between rooms: adjacency (*adj*) and accessibility



**Fig. 6.** a) The final project of the attic, b) the hierarchical layout hypergraph representing the divided area

(acc). The first one is denoted by the continuous line, while another by the dashed one.

Then, the hypergraph  $h_1$  is nested in the hyperedge of  $h_0$  (Fig.2a). The resulting hypergraph  $h_2$  is shown Fig.2b. As the result of the nesting operation the nodes of the initial hypergraph were substituted by corresponding external nodes of the nested hypergraph. The labels of new external nodes are a concatenation of the initial hypergraph node labels with labels of the nodes of  $h_1$ . The node 1 of  $h_0$  is replaced by the node 1.1, the nodes 2 and 4 are replaced by nodes 2.2 and 8.4, respectively. The node 3 is replaced by three following nodes: 3.3 and 7.3 which represent two walls of the living area and the node 3.3 which represents a wall of the landing. Although there are two nodes labelled as 3.3, they can be unambiguously identified as they are numbered differently as

external nodes of  $h_2$ . Nodes 4, 5, 6 of the living area and nodes 1, 2 and 4 of the landing are internal nodes of the hypergraph  $h_2$ .

In the next step the designer divides the living area into two separate flats (Fig.3a). The resulting hypergraph contains a hierarchical node which represents a wall obtained by connecting two collinear walls of flat  $M1$  corresponding to nested nodes 2 and 3.6. Then the designer creates the layout of both flats. The flat  $M1$  is divided into the living room, kitchen, bathroom and hall, while the flat  $M2$  is divided into the living room, sleeping room, kitchen and bathroom (Fig.4a). The resulting hypergraph is shown in Fig.4b.

Let us assume that the designer is not satisfied with the layout of the flat  $M2$  and he decides to change it. Therefore he removes the division of the area  $M2$  in the diagram (Fig.5a). As a consequence the hyperedge suppression operation on the layout hypergraph is invoked automatically. As the result of this operation the layout hypergraph nested in the hyperedge labelled  $M2$  is removed. The obtained hierarchical layout hypergraph is shown in Fig.5b. The nodes which were divided by the development operation are merged again. For example two nodes with labels 1.1.1.1 representing north walls of the sleeping room and leaving room, respectively, are merged into one node with label 1.1.1 representing the north side of the area  $M2$ . The design diagram which is obtained as a result of a new division of  $M2$  and the corresponding hierarchical layout hypergraph are presented in Fig.6.

## 5 Conclusions

The next problem of the knowledge-based decision support system is connected with reasoning about design solution features and suggesting design modifications. To this end we intend to equip hypergraphs with attributes reflecting characteristics of designs. The semantic design knowledge stored in an attributed hypergraph representation will allow for reasoning about layouts.

## References

1. Grabska E, Grzesiak-Kopeć K, Lembas J, Lachwa A, Ślusarczyk G (2006) Hypergraphs in Diagrammatic Design. Computer Vision and Graphics, Proceedings of the International Conference, ICCVG 2004, Springer, 111–117
2. Minas M (2002) Concepts and Realization of a Diagram Editor Generator Based on Hypergraph Transformation, Science of Computer Programming 44:157–180

---

# Articulation Rate Recognition by Using Artificial Neural Networks

Izabela Szczurowska<sup>1</sup>, Wiesława Kuniszyk-Józkowiak<sup>2</sup>, and Elżbieta Smółka<sup>2</sup>

<sup>1</sup> Department of Physics, Agricultural University, Akademicka 13, 20-950 Lublin, Poland

[izabela.szczurowska@ar.lublin.pl](mailto:izabela.szczurowska@ar.lublin.pl)

<sup>2</sup> Laboratory of Biocybernetics, Institute of Computer Science, Maria Curie-Skłodowska University, pl. Maria Curie-Skłodowska 1, 20-031 Lublin, Poland

[natalka@tytan.umcs.lublin.pl](mailto:natalka@tytan.umcs.lublin.pl), [esmolka@tytan.umcs.lublin.pl](mailto:esmolka@tytan.umcs.lublin.pl)

**Summary.** This work concerns the problem of the application of artificial neural networks in the modelling of the hearing process. The aim of the research was to answer the question whether artificial neural networks are able to evaluate speech rate. Speech samples, first recorded during reading of a story with normal and next with slow articulation rate were used as research material. The experiment proceeded in two phases. In the first stage Kohonen network was used. The purpose of that network was to reduce the dimensions of the vector describing the input signals and to obtain the amplitude-time relationship. As a result of the analysis, an output matrix consisting of the neurons winning in a particular time frame was received. The matrix was taken as input for the following networks in the second phase of the experiment. Various types of artificial neural networks were examined with respect to their ability to classify correctly utterances with different speech rates into two groups. Good examination results were accomplished and classification correctness exceeded 88%.

## 1 Introduction

Speech rate and other coefficients such as the pitch, the formants, the mel-frequency cepstral coefficients, the intensity of speech signal etc, are features mainly used in speaker recognition. Speakers are alternately speeding up and slowing down speech production, using pauses and using variations in speech tempo [1] causing changes both in spectral characteristic and in word pronunciation that affect automatic speech recognition systems [2]. Although articulation rate allows for identifying a particular speaker, it is one of the major factors in emotion recognition [3]. Rate of speech is interpreted as the rate at which individual speech elements are uttered. The universal method of speech rate measurement does not exist. Articulation rate measures vary in the choice of the speech units that are used in calculation for example phone rate, syllable or word rate [4]. Additionally, rate of speech is one of the features taken into account in non-fluent speech diagnosis, prediction, and in creation of disfluency

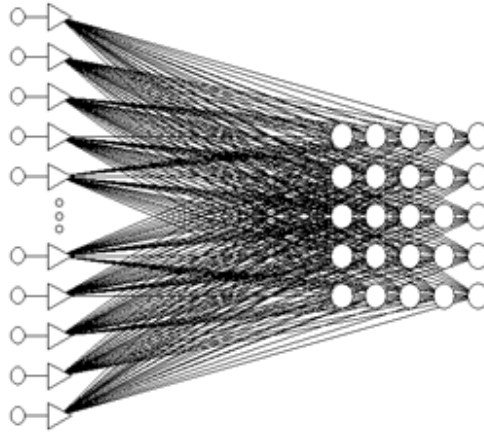
recognition systems [5, 6]. At present, neural networks play an important role both in speech [7] and in speaker recognition [8] and are an irreplaceable tool in the case of the necessity to distinguish between very similar signals. The authors of the following article applied artificial neural networks to recognize and classify utterances with normal and slowed rate of speech. The first stage of the hearing process projection was an acoustical examination where spectral analysis was used in order to recreate in approximate way the processes which take place in human ear. Next Kohonen network (SOM) was applied, which allows for obtaining the amplitude-time relationship for speech signals. The former stage could be compared to data collection by neurons in the inner ear and in cochlea nerve. The latter phase, which was equivalent to the processes taking place in the brain, consisted in speech classification and recognition. It was realized by two various types of neural networks: MLP and RBF.

## 2 Research Materials

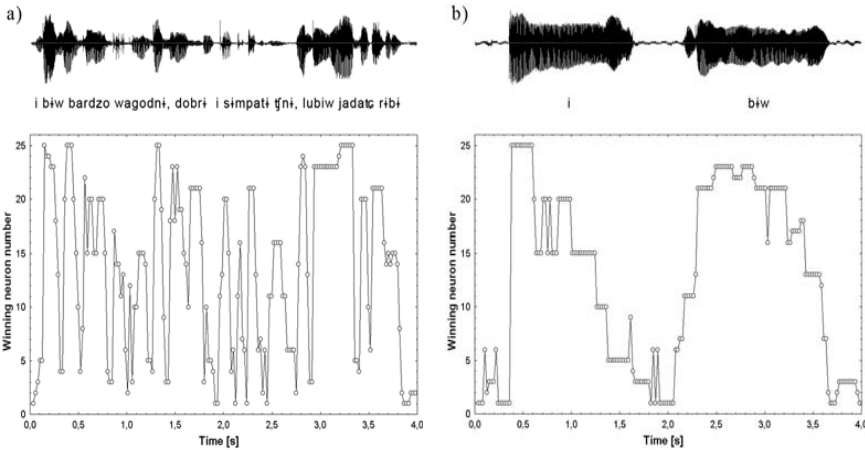
Five people participated in the experiment. They read a short story twice: first at a normal rate and then at a slow rate. The slowing-down of the rate was forced by the application of a delayed auditory feedback whose delay time equaled 200ms. The research material was recorded in a sound booth with the use of a Sound-Blaster card controlled by the Creative Wave Studio application. The signal was transformed from an analogue form into digital with the sampling frequency of 22,050 Hz and sampling precision of 16 bits. From among the recorded files 50 four-second utterances at a normal rate and 50 utterances at a slow rate were chosen. The 4-second fragments were selected from the recorded utterances with the use of the Creative Wave Studio, which allows us to observe the speech signal in the form of an oscillogram and to listen to it simultaneously. The samples were analysed by FFT 512 with the use of the SpectraLAB application with the time resolution of 23ms. A-weighting filter and digital 1/3-octave filters were applied. A-weighting filter makes it possible to transform a speech signal, including the specific features of human hearing. The examination with digital 1/3-octave filters reflects the physical sense of the frequency analysis, which is carried out on a basement membrane in the inner ear. In that experiment, 21 digital 1/3-octave filters of centre frequencies between 100 and 10,000 Hz were used. Resulting from the analysis, 100 utterances were obtained. The parameters of the utterances were similar to the signal which goes from the inner ear to the brain. These samples were used as input for SOM network.

## 3 First Artificial Neural Network Layer

The first network, a Self-Organizing Map, was used to reduce the dimensions describing the input signals. The input data of the SOM were vectors obtained in the former analysis. The influence of the structure and learning parameters on the network working abilities was checked. The analysed networks had a



**Fig. 1.** SOM with 171 inputs and 25 neurons in output layer



**Fig. 2.** Utterance with normal (a) and delayed (b) speech rate representation with corresponding oscillograms

rectangular shape, since that structure permits us to project and visualize high-dimensional signal into two-dimensional space [9] and consisted of different numbers of output neurons. The results of their work were evaluated for various learning parameters. Since the only way to assess the work of the network was to define the correctness of description of the examined samples on the basis of the syllabic structure reflection of the sample on the chart of the winning neurons' relation to time, the required analysis was conducted. Finally, Kohonen network, built with 21 inputs and 25 neurons in the output layer (Fig. 1), was taught in 100 epochs, with the stable learning rate which equalled 0.1 and with neighbourhood decreasing from 3 to 0.



The application of the network caused a notable reduction of the dimensions describing the examined signals as well as made it possible to model the syllabic structure of utterances and obtain the amplitude-time relationship for speech signals. All the samples, both with normal and delayed speech rate, were represented by figures showing the dependence of the winning neurons on time (Fig. 2).

## 4 Second Neural Network Layer

The result of previous analysis was a chart consisting of winning neuron numbers in a particular time frame. That chart was used during the second network training. It consisted of 100 four-second utterances, which were divided into three classes: teaching, verifying and testing. The aim of that network layer, where the speech rate perception process occurred, was to group samples into two groups. MLP and RBF networks were used and next evaluated as classifiers. Those networks were supposed to act as a part of the nervous system, assessing the examined feature.

MLP and RBF networks had 171 neurons in the input layer due to receiving 171 time points for each sample, and 1 neuron in the output because the answer "normal" or "low" (which means that an output neuron was activated or not) was expected. The architecture and learning parameters for each network were chosen in the experimental way.

**Table 1.** Quality and teaching error statement for teaching, verifying and testing data for table with 100 samples

Network	Teaching quality	Validation quality	Testing quality	Teaching error	Validation error	Testing error
MLP	0.69	0.51	0.49	0.76	1.15	1.11
RBF	0.64	0.51	0.46	0.43	0.49	0.54

Due to the fact that the networks did not achieve acceptable results (Tab. 1), sensitivity analysis was applied and showed that 4-seconds utterances were too long. In relation to that, the speech samples were cut into 1-second fragments and taken as 400 individual utterances (200 with normal and 200 with slow speech rate). As a result of the undertaken input data transformations, MLP and RBF networks finally had 42 inputs (one-second utterance) and, as in the previous case, 1 output neuron. MLP had two hidden layers, both built with 20 neurons and was taught using Back Propagation algorithm in 100 epochs and with Conjugate Gradient algorithm in 500 epochs, with the learning rate decreasing from 0.7 to 0.1 and 0.3 momentum. The logistic function was used as an activation function. Error was calculated using cross-entropy error function, which is used in particular with classifying networks. RBF hidden layer consisted of 92 neurons.

In that case, sampling (drawing of random samples) and K-nearest neighbourhood algorithms were applied. Error was also cross-entropy error function. All networks were assessed using the following test precision parameters based on the error matrix: accuracy, sensitivity and specificity [10, 11]. In the conducted calculations, sensitivity and specificity changes during displacing their boundary value were represented by ROC (Receiver Operating Characteristic) curves.

## 5 Results

The extension of teaching data had a positive influence on the obtained results. Both MLP and RBF managed to classify correctly more than 70% of utterances (Tab. 2).

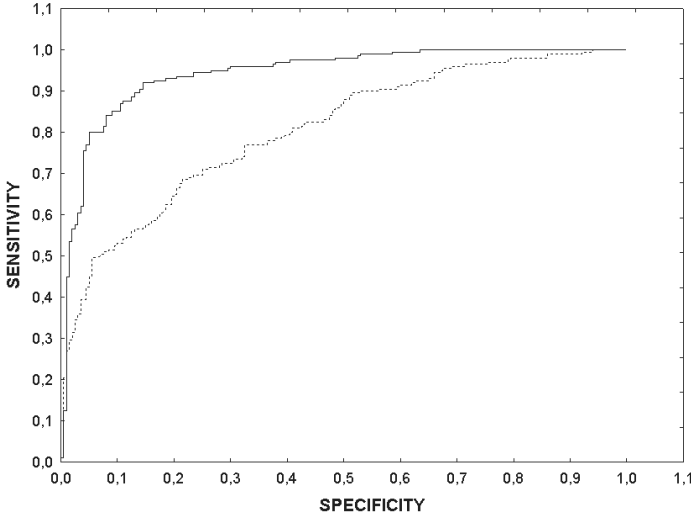
**Table 2.** Quality and teaching error statement for teaching, verifying and testing data for table with 100 samples

Network	Teaching quality	Validation quality	Testing quality	Teaching error	Validation error	Testing error
MLP	0.80	0.70	0.70	0.64	0.67	0.71
RBF	0.95	0.84	0.78	0.25	0.35	0.36

**Table 3.** Evaluation coefficients values for MLP and RBF networks

Network Speech rate	RBF		MLP	
	normal	slow	normal	slow
All	200	200	200	200
Correctly classified	177	175	152	155
Incorrectly classified	23	25	48	45
Unknown	0	0	0	0
Classification correctness (%)	88.5	87.5	76	77.5
Classification error (%)	11.5	12.5	24	22.5
Unknown (%)	0	0	0	0

Correctness of the studied networks could be considered with respect to error made by each network during teaching, verifying and testing, which, in the majority, informs about network’s appropriate performance. The errors made by MLP during classification, in teaching and verifying groups were close (respectively 0.64 and 0.67) and higher than in residual cases (Tab. 2). Testing error was slightly higher in this case, however, it did not differ to a large extent from others, which showed that the network is able to generalize. RBF made some mistakes in teaching and verifying data; additionally error during testing was significantly lower than in MLP case. Data set error level was rather stable, thus, as in the previous example, the network was able to generalize input data correctly.



**Fig. 3.** ROC curves for RBF (continuous line) and MLP (broken line) networks

**Table 4.** Evaluation coefficients values for MLP and RBF networks

Network	Accuracy	Sensitivity	Specificity
MLP	0.76	0.76	0.78
RBF	0.88	0.89	0.88

If we take into consideration the number of mistakes which were made by the networks, the best was RBF. Network incorrectly classified 48 utterances, 23 of which had been uttered with normal and 25 with slow speech rate, which brings 88% of the categorizing correctness (Tab. 3). RBF achieved accuracy, sensitivity and specificity respectively at the levels: 0.88, 0.89, 0.88 (Tab. 4). Additionally, the area under the ROC curve totalled 0.94 (Fig. 3), which means, that this classifier achieved impressive results.

MLP network incorrectly allocated 93 utterances (48 with normal and 45 with slow speech rate) - classification correctness was about 76.75% (Tab. 3) while area below ROC curve 0.80 (Fig.3). MLP accuracy was 0.76, sensitivity 0.76 and specificity 0.78 (Tab. 4).

## 6 Conclusion

Our research has shown that artificial neural networks can be a useful tool in speech analysis. The application of the first network has allowed us to reduce the dimensions describing the input signals, obtain the amplitude-time relationship for a speech signal and made it possible to claim that Kohonen network can

be used in describing speech. As it can be observed, Kohonen network reflects the syllabic structure of utterances very well. Next, the networks were examined for correct recognition and classification of samples into two groups. The best RBF network managed to classify correctly 88% of the samples. The values of the obtained test precision parameters - accuracy (acc), sensitivity (sensi) and specificity (speci) - point to the good quality of the examined classifier and encourage its further development.

## References

1. Koopmans-van Beinum FJ, van Donzel ME (1996) Relationship between discourse structure and dynamic speech rate. In: Bunnell HT, Idsardi W (Eds), Proceedings ICSLP96, Vol 3, 1724-1727
2. Zheng J, Franco H, Stolcke A (2000) Rate-dependent acoustic modeling for large vocabulary conversational speech recognition. In: Proceeding ISCA Tutorial and Research Workshop on Automatic Speech Recognition: Challenges for the new Millennium, Paris, France, 145-149
3. Ververidis D, Kotropoulos C (2006) Emotional speech recognition: Resources, features and methods. *Speech Communication* 48, 1162-1181
4. Verhasselt JP, Martens JP (1996) A fast and reliable rate of speech detector, Proceedings ICSLP96, Vol 3, 2258-2261
5. Guitar B, Kopff-Schaefer H, Donahu-Kilburg G, Bond L (1992) Parent verbal interactions and speech rate: A case study in stuttering. *Journal of Speech and Hearing Research* 35, 742-754
6. Howell P, Sackin S (2000) Speech rate modification and its effects on fluency reversal in fluent speakers and people who stutter. *Journal of Developmental and Physical Disabilities* 12(4),
7. Chen WY, Chen SH, Lin CHJ (1996) A speech recognition method based on the sequential Multi-layer Perceptrons. *Neural Networks* 9(4), 655-669
8. Farrell K, Mamione R, Assaleh K (1994) Speaker recognition using neural networks and conventional classifiers. *IEEE Transaction on Speech and Audio Processing* 2(1), part 2, 194-205
9. Kohonen T (2001) *Self-Organizing Maps*. Springer, Berlin, Heidelberg, New York
10. Kestler HA, Schwenker F (2000) Classification of high-resolution ECG signals, In Howlett R, Jain L *Radial basis function neural networks: theory and applications*. Heidelberg: Physica-Verlag
11. Schwenker F, Kestler HA, Palm G (2001) Three learning phases for radial-basis-function networks. *Neural Networks* 14, 439-458.

---

# Recognition of Cycles of Repeated Hypoxia on the Basis of Time Periods in Biological Model

Beata Sokolowska<sup>1</sup> and Adam Jozwik<sup>2</sup>

<sup>1</sup> Department of Respiratory Research, Medical Research Center, Polish Academy of Sciences, Pawińskiego 5, PL02-106 Warsaw, Poland

sokolowskab@cmdik.pan.pl

<sup>2</sup> Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences, Trojdena 4, PL02-109 Warsaw, Poland

adamj@ibib.waw.pl

**Summary.** Recognition possibility of expositions (cycles) on the basis the respiratory response to repetitive hypoxia in experimental model (named as the intermittent hypoxia model) was the aim of the study. Variables of a breathing pattern: respiratory frequency, tidal volume, duration of inspiration and expiration ( $F$ ,  $V_T$ ,  $T_I$ ,  $T_E$ , respectively), and phrenic nerve activity ( $A_{phr}$ ) were measured in one-minute periods in biological experiments. The data set was analyzed using  $k$ -nearest neighbor decision rule ( $k$ -NN). It was found that the variables, measured after sequential hypoxic exposures for the recovery phases, allow recognizing cycles with 6.5% of misclassification rate. However, the recognition based on two selected variables  $\{F, V_T\}$  offered significantly lower error rate of 2%. The obtained results have shown the usefulness of pattern recognition approach to diagnosis of the changes that reflect the respiratory plasticity developing on short-term exposures of the intermittent hypoxia, such as that simulated in the biological model.

## 1 Introduction

Repeated hypoxic episodes are frequently observed in our life and in different pathological situations, e.g. in human subjects during obstructive and central apnea particularly during sleep at high altitude [5], [8]. Experimentally, this phenomenon is defined as the intermittent hypoxia (IH). Two kind of phases occurs in IH: repeated exposures of hypoxic stimulus, named as a stimulus phase ( $S$ ), which are interspersed with episodes of normoxia or hyperoxia, termed as a recovery phase ( $Rec$ ). Such successive hypoxic-reoxygenation episodes, consisted of  $S$  and  $Rec$ , are defined as cycles. It has been shown that chronicity of cyclic hypoxic exposures has an influence on elicited hypoxic ventilatory responses (HVR) to the acute hypoxia and on ventilatory long-term facilitation (LTF), in animal models of the respiratory diseases. Recent data point to a key role of reactive oxygen properties in the development of respiratory plasticity in chronic intermittent hypoxia-evoked augmentation of LTF [6], [7]. Carotid chemoreceptors

also respond to the cyclic hypoxia [11]. It seems that the function of neural serotonin receptors also determines LTF as their blockade in the spinal cord abolishes LTF [1], [2]. Serotonergic pathways are the critical mediator of this form of neural plasticity in respiratory system. Furthermore, the neurotrophic factor BDNF (brain derived neurotrophic factor) is important because it implicates the initiation and consolidation of LTF phenomenon [3]. It is worth to understand the appearing changes and mechanisms underlying IH-induced ventilatory plasticity, if one is interested in developing of supporting and therapeutic methods. In the subchronic form, trainings of cyclic periods of hypoxia are part of a variety of intermittent hypoxic exercises use in recreative sports (mountaineers, bicyclists, and runners), in aircraft and cosmonautics. Moreover, these approaches are used as therapeutic methods in sport medicine and clinic of system's disorders [4], [9], [12]. There are following reasons of studying IH: to evaluate its destructive factor, to propose suitable treatment of diseases and to propose suitable training for healthy subjects. Statistical methods are considered as basis of computer diagnosis system for recognition of observed early respiratory changes in animal model with repeated hypoxia. The results might be applied in training protocols of sport physiology or therapeutic programs used in diseases.

## 2 Materials and Methods

### 2.1 Biological Experiments and Measurement of Respiratory Variables

The biological experiments were performed on 8 adult's male rabbits, which were anesthetized and spontaneously breathing. These experiments were characterized as an experimental set  $Ex = \{Ex1, Ex2, Ex3, Ex4, Ex5, Ex6, Ex7, Ex8\} = \{Ex(i)\}$ , where  $i=1, 2, \dots, 8$  is number of the experiment. Detailed description of biological procedure was presented in the previous paper (e.g. [10]). Variables of a breathing pattern: respiratory frequency ( $F$ ), tidal volume ( $V_T$ ), duration of inspiration and expiration ( $T_I$  and  $T_E$ , respectively), were recorded in the experiments. The output of the respiratory centre or neural drive is the best expressed in the phrenic neurogram, so a signal of phrenic nerve activity ( $A_{phr}$ ) was also recorded, and maximal amplitude of the integrated activity of the nerve was calculated. Finally, in each independent experiment multidimensional variable vectors consisting of five components  $Var(Ex(i)) = \{F(Ex(i)), V_T(Ex(i)), T_I(Ex(i)), T_E(Ex(i)), A_{phr}(Ex(i))\}$ ,  $i=1, 2, \dots, 8$ , were observed. They were used in the statistical analysis. The experiments obtained approval of local Ethics Committee.

### 2.2 The Repeated Hypoxia Procedure (The Intermittent Hypoxia Model, IH)

Researchers used different protocols of repeated hypoxia (e.g. [5]). In our experiments, we chose a medium duration of hypoxic episodes of 1-min and moderate intensity of hypoxia of 14%, so animals breathed hypoxic mixtures of 14% O<sub>2</sub>

in N<sub>2</sub> in the stimulus phase. The normoxic recovery phase lasted 3-min, and was divided in three 1-min periods. So, the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> minute of the corresponding normoxia phase, *Rec1*, *Rec2*, *Rec3*, was considered. We applied five-trial hypoxic (stimulus) - normoxic (recovery) of cyclic repeated hypoxia, and in each minute the measurements of five breaths were recorded. The time-conditions of the intermittent hypoxia are shown on fig. 1. Thus, the exposures consisted of 5 successive hypoxic-normoxic cycles. Each cycle of IH was determined by five following minutes: (1) before hypoxia (*B period*, before given an actual stimulus), (2) during hypoxia (*S period*, it is the stimulus phase), (3) 1<sup>st</sup> minute recovery (*Rec1 period*), (4) 2<sup>nd</sup> minute recovery (*Rec2 period*), (5) 3<sup>rd</sup> minute recovery (*Rec3 period*), i.e. there were 3 intervals of the normoxic recovery phase. Measurement vectors of all 1-min periods,  $P(Ex(i)) = \{B(Ex(i)), S(Ex(i)), Rec1(Ex(i)), Rec2(Ex(i)), Rec3(Ex(i))\}$ , where  $i=1, 2, \dots, 8$  is number of the experiment, were gathered for each of 5 variables.

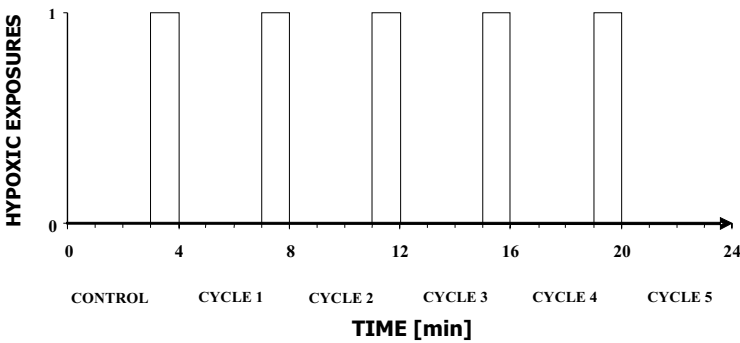


Fig. 1. The short-term intermittent hypoxia model

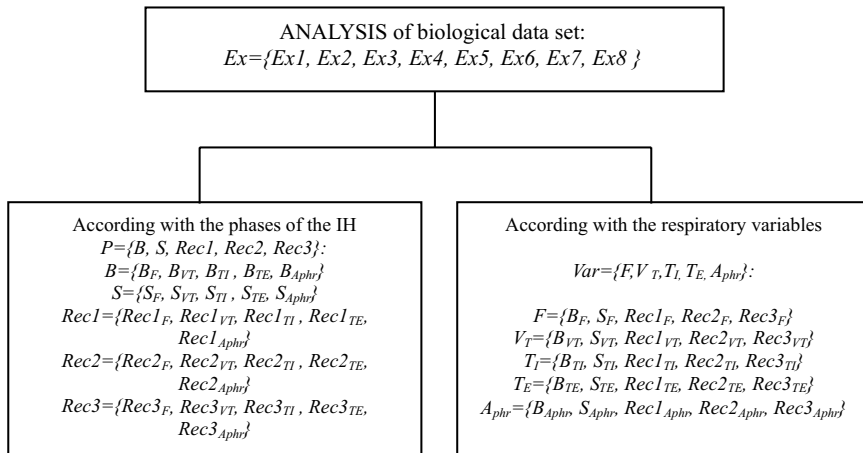
**2.3 Statistical Methodology**

The values of variables of breathing pattern and phrenic nerve activity were considered as the experimental data set. A possibility of recognition the cycles (treated as classes) during successive episodes of repeated hypoxia according with one-minute periods (see previous paragraph) was analyzed.

Two questions - important from a point of view of the respiratory biology are interesting: (1) which of one-minute periods differentiate the most the considered hypoxic-normoxic cycles of IH, (2) which combination of the variables is most informative to recognize the cycles. To answer these questions the features were gathered in two manners: according to the one-minute periods (fig. 2, left bottom square) and in accordance with the single variable (fig. 2, right bottom square). To answer these questions the methods of statistical pattern recognition were be applied.

The data analysis was performed with the use of pair-wise classifier based on the *k*-*NN* rule and the leave one out method. The *k*-*NN* rule can be used for arbitrary number of classes, but decomposition of multiclass decision problem into

two decision ones offers usually lower error rate, i.e. misclassification percentages. The pair-wise classifier is a multiclass classifier that consists of two-class  $k$ -NN classifiers. The final decision of the pair-wise classifier is formed by voting of all component classifiers. Numbers  $k$  are being established experimentally separately for each of the component classifiers.



**Fig. 2.** Scheme of analysis of the experimental intermittent hypoxia (IH), description in the text

Some features available in the reference set can be redundant or not related with the considered classes and they may cause increasing of the error rate. So, feature selection is recommended. Determination of  $k$  nearest neighbors and feature selection requires error rate estimation for different  $k$  and different feature combinations for each of the component classifiers. The leave one out method was chosen for an error rate calculation.

### 3 Results and Discussion

Every out of 8 experiments was analyzed individually, and then the analysis was done for all experiments together. The analysis was carried without and with selection of features (vectors) for 5 cycles and also for the two extreme cycles, i.e. for the first and the last cycle of repeated hypoxia. These results are described below.

In *experiment no.1*, the great misclassification rates ( $E_r$ ) with range of 32%-52% were received in 1-min minute periods. After selection of features the rate of mistakes decreased to 4%-28%. Two vectors, frequency  $F$  and tidal volume  $V_T$ , offered the lowest error rate that was equal 8%. The extreme cycles, the first and the last, were recognised correctly by all variables except for inspiration time  $T_I$ .



The global error rate  $E_r$  for recognizing cycles without selection of features was equal 12%, and the extreme cycles were correctly recognised.

*In experiment no.2*, like as in experiment 1, large rates of mistakes with range of 20%-36% were received. After selection of features the mistakes decreased to 4%-12%, and the extreme cycles 1<sup>st</sup> and 5<sup>th</sup> were recognized with 0%-10% of mistakes. The lowest misclassification rates were received for the 2<sup>nd</sup> and the 3<sup>rd</sup> minute of the recovery phase. The vectors  $F$ ,  $V_T$ ,  $T_E$  and  $A^{phr}$  gave the lowest  $E_r$  for recognizing 5 cycles, what was established after selection of features. The cycles 1<sup>st</sup> and 5<sup>th</sup> were recognized quiet correctly ( $E_r=0\%$ ), except for inspiration duration and activity of phrenic nerve ( $E_r(T_{I(1,5)})=E_r(A_{phr(1,5)})=10\%$ ). The global misclassification rate for 5 cycles without feature selection as well as for the extreme cycles was equal 0%.

*In experiment no.3*, the classifier gave the good recognition quality except the stimulus phase ( $S$ ). Feature selection in this experiment permitted the correct recognizing of all cycles for every minute of intermittent hypoxia. The three ventilatory components  $F$ ,  $V_T$  and  $T_E$  enabled correct classification of 5 classes.

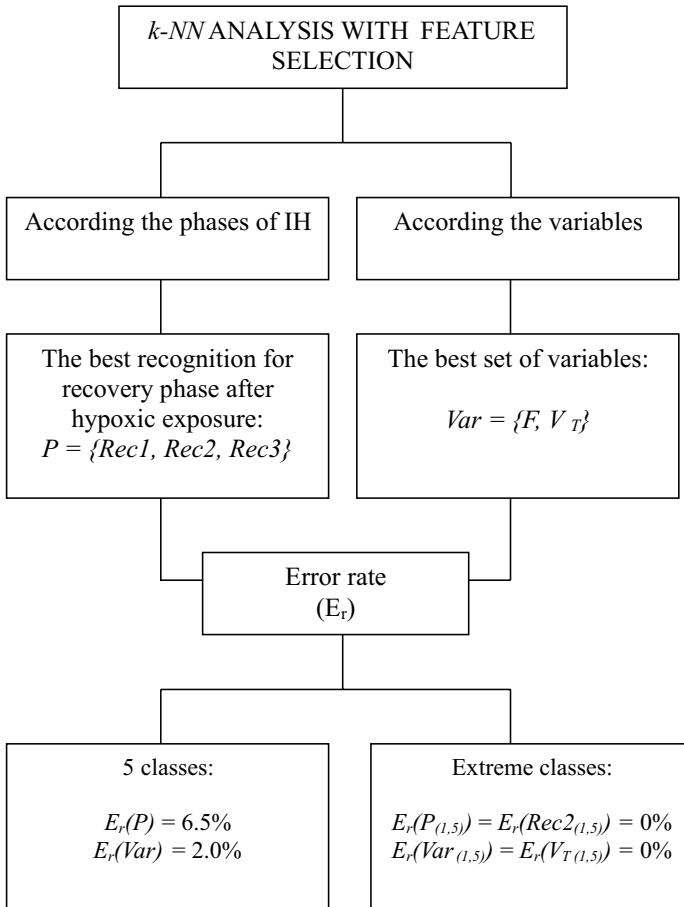
*In experiment no.4*, the error rates were equal from 4% to 36%, and after feature selection they decreased to 4%-16%. The most effective in cycle recognition was the 1<sup>st</sup> minute of the recovery phase. The recognition between cycle 1<sup>st</sup> and 5<sup>th</sup> was quite correct. It was established that each of the vectors  $F$ ,  $V_T$  and  $A_{phr}$  also offered the zero misclassification rates. The global  $E_r$  without feature selection was equal 0%.

*In experiment no.5*, the all cycles were recognized with great error rates of 32%-60%, which decreased to half of their values, when feature selection was done. The extreme cycles, after selection of features, were correctly recognised. The  $Rec2$  of recovery phase was the most effective in cycle recognizing. None of the considered respiratory vectors gave the correct classification of the repeated hypoxia cycles. However, it was possible to distinguish correctly the first and the last cycles with the use of vectors  $F$ ,  $V_T$  and  $A_{phr}$  (even without feature selection). The global error rate was equal 24% and it was the greatest (without feature selection) among all experiments, however extreme cycles were recognised correctly.

*In experiment no.6*, the  $E_r$  for recognizing 5 cycles were equal between 4% and 16%. After feature selection, the 5 cycles during stimulus as well as in recovery phases were differentiated correctly. The extreme cycles were recognised correctly in every 1-minute periods. The vectors  $F$ ,  $V_T$  and  $T_E$  allowed the correct recognition of 5 cycles. The cycles 1<sup>st</sup> and 5<sup>th</sup> were correctly recognised with exclusion of  $T_I$ . The global error rate was 0%.

*In experiment no.7*, the rate of wrong classification of 5 cycles varied between 20% and 44%. However, using only  $Rec3$  period this error rate decreased to zero. The extreme cycles were correctly recognised in all cycle minutes except the stimulus phase ( $E_r(S_{(1,5)})=10\%$  after the feature selection). The global  $E_r$  was equal 0%.

*In experiment no.8*, the cycles of IH weren't recognised correctly in no phases; however first cycle and last one were distinguished well. The vector  $F$  permitted



**Fig. 3.** Results of *k-NN* analysis with feature selection in biological model of the intermittent hypoxia

the correct recognizing of 5 cycles. After feature selection, the vectors  $F$ ,  $V_T$  and  $T_E$  also allowed the correct recognize cycles. The extreme cycles were recognized correctly by the vectors  $F$ ,  $V_T$ ,  $T_E$ , but not by  $T_I$ . The global  $E_r$  was equal 0%.

The analysis of the data gathered jointly from all the eight experiments indicated the great error rates for every minute of intermittent hypoxia, i.e. for the stimulus as well as for recovery phases. The  $E_r$  for recognizing 5 cycles without feature selection were of the range 32%-39%, and after feature selection they were 23%-30%. Differentiation between the first cycle and the last one was better in the recovery phase with 2.5%-6.2% in comparison with the stimulus phase with 12.5% of errors. These mistakes were decreased almost twice after feature selection. The recognition between 1<sup>st</sup> and 5<sup>th</sup> cycles was faultless for *Rec2*

recovery period and tidal volume  $V_T$  in all single experiments. The vectors  $F$  and  $V_T$  gave the lowest error rates for recognizing 5 cycles. Both these vectors ( $F$  or  $V_T$ ) and expiration time ( $T_E$ ) offered the most effective recognizing the cycles of 1<sup>st</sup> and 5<sup>th</sup> ( $E_r(F_{(1,5)})=6.2\%$ ,  $E_r(V_{T(1,5)})=2.5\%$  and  $E_r(T_{E(1,5)})=6.2\%$ , without feature selection). The global misclassification rate for recognizing 5 cycles, without feature selection, was equal scarcely 11%, and the extreme cycles were recognised perfectly. When only a recovery phase of IH was analysed (represented by vector  $\{Rec1, Rec2, Rec3\}$ ), then the error rate for recognizing 5 cycles decreased to 6.5% (fig. 3). The use of two components of the pulmonary ventilation  $\{F, V_T\}$ , diminished the misclassification rate even to 2%, as it is presented on fig. 3. The global error rate for 6 out of 8 experiments equaled to 0% for the all 5 classes, but for the extreme cycles this error rate always was equal 0%.

Analysis of the ventilatory components  $\{F, V_T\}$  and the recovery phase  $\{Rec1, Rec2, Rec3\}$  allowed for the best recognition of early respiratory effects on repeated hypoxic exposures. The time of expiration ( $T_E$ ) enabled the better recognition of the cycles than the time of inspiration ( $T_I$ ). The activity of phrenic nerve ( $A_{phr}$ ) has given worse recognition than the ventilation components and breathing times. Faultless recognition of extreme cycles points on different breathing states as an effect of functional plasticity abilities of respiratory system.

*In conclusion*, the performed analysis of data indicates on possibility of diagnosis of observed changes that reflect plasticity of respiratory output developing on short-term cycles of hypoxia. This approach may be applied in preparation of training protocols in sport physiology and therapeutic program used to compensate for respiratory pathologies.

**Acknowledgement.** His work was supported by the statutory budget of Medical Research Center and Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences. The authors are thankful to Mrs. Ewa Wi-  
elechowska for excellent assistance of biological experiments.

## References

1. Golder FJ, Mitchell GS (2005) Spinal synaptic enhancement with acute intermittent hypoxia improves respiratory function after chronic cervical spinal cord injury. *J Neuroscience*, 25(11):232
2. McGuire M, Zhang Y, White DP, Ling L (2004) Serotonin receptor subtypes required for ventilatory long-term facilitation and its enhancement after chronic intermittent hypoxia in awake rats. *Am J Physiol Regul Integr Comp Physiol*, 286: R334-R341
3. Mitchell GS, Backer TL, Nanda SA, Fuller DD, Zabka AG, Hodgeman BA, Bavis RW, Mack KJ, Olson EB (2001) Invited review: Intermittent hypoxia and respiratory plasticity. *J Appl Physiol*, 90:2466-2475
4. Morton JP, Cable NT (2005) Effects of intermittent hypoxic training on aerobic and anaerobic performance. *Ergonomics*, 48: 1535-1546

5. Neubauer JA (2001) Invited Review: Physiological and pathophysiological responses to intermittent hypoxia. *J Appl Physiol*, 90: 1593-1599
6. Peng Y-J, Prabhakar NR (2003) Reactive oxygen species in the plasticity of respiratory behavior elicited by chronic intermittent hypoxia. *J Appl Physiol*, 94:2342-2349
7. Prabhakar NR (2001) Invited Review: Oxygen sensing during intermittent hypoxia: cellular and molecular mechanism. *J Appl Physiol*, 90: 1986-1994
8. Prabhakar NR, Fields RD, Baker T, Fletcher EC (2001) Intermittent hypoxia: cell to system. *Am J Physiol*, L524-L528
9. Roels B, Millet GP, Marcoux CJ, Coste O, Bentley DJ, Candau RB (2005) Effects of hypoxic interval training on cycling performance. *Med Sci Sports Exerc*, 37: 138-146
10. Sokolowska B, Jozwik A (2005) Statistical evaluation of ventilatory patterns in response to intermittent hypoxia in the rabbit. *J Physiol Pharmacol*, Suppl4, 56: 203-207
11. Sokolowska B, Pokorski M (2006) Ventilatory augmentation by acute intermittent hypoxia in the rabbit. *J Physiol Pharmacol*, Suppl4, 57: 341-347
12. Srebrovskaya TV, Swanson RJ, Kolesnikova EE (2003) Intermittent hypoxia: mechanisms of action and some applications to bronchial asthma treatment. *J Physiol Pharmacol*, 54 Suppl1: 35-41

---

# Fingerprint Reference Point Detection Using Neighbourhood Influence Method

Piotr Porwik<sup>1</sup> and Lukasz Wieclaw<sup>2</sup>

<sup>1</sup> University of Silesia, ul. Bedzinska 39, 41-200 Sosnowiec, Poland  
porwik@us.edu.pl

<sup>2</sup> University of Silesia, ul. Bedzinska 39, 41-200 Sosnowiec, Poland  
wieclaw@netfactor.pl

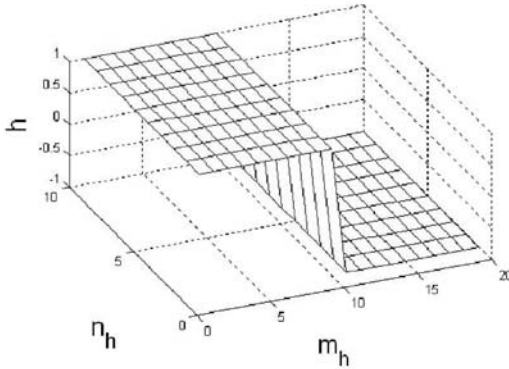
**Summary.** The uniqueness of a fingerprint allows to build systems that can recognize personal fingerprint feature automatically. In practice, fingerprint recognition and classification based on two techniques: the reference point location and the Gabor filtering or the fingerprint ridge characteristics (called minutiae). In this paper a new method of the reference point alignment has been presented. A new approach of reference point localization is based on so-called identification filters, which have been composed on the basis of analysis of biometric characteristic of human fingerprint. In this paper construction of such filters has been precisely explained. Proposed method with other methods was compared.

## 1 Introduction

A fingerprint is the reproduction of a fingertip epidermis, produced when a finger is pressed against a smooth surface. The most evident characteristic of a fingerprint is a pattern of interleaved ridges and valleys in image of fingerprint. Ridges are dark whereas valleys are bright (see 1a). Fingerprint identification is one of the most important biometric technologies especially in Fingerprint Identification System (AFIS). A fingerprint is the pattern of ridges and valleys on the surface on a fingerprint. Each human has unique fingerprints. The uniqueness of a fingerprint allows to build systems, which can recognize personal fingerprint feature automatically. Unfortunately it is a difficult task, because fingerprint images have very poor quality very often. In practice, due to variations in impression conditions, ridges configuration, skin condition, acquisition devices, etc. many restrictions have to be considered. The ridge structures in poor-quality fingerprint images are not always well defined, therefore they cannot be detected. Fortunately a fingerprint experts are often able to correct such corrupted images as long as the ridges and valleys structure is not corrupted completely. Each analyzed digital fingerprint image should have area, where structure of finger will be visible. Generally, fingerprint images can be divided into three types: well-defined region, recoverable corrupted region and unrecoverable region [1]. In the first region ridges and valley are clearly differentiated, in the second region ridges

and valleys are corrupted by a small amount of creases or smudges but are still visible. In the third area ridges and valleys are strong corrupted - such image is rejected. In our paper only two first areas are analyzed and each of them can be used to fingerprint recognition.

## 2 Estimation of Local Ridges Orientation



**Fig. 1.** The values of the mask  $h$  for the  $\lambda = 0^\circ$  angle

An orientation field represents local orientations of fingerprint ridges, which occur on fingerprint surface. This field represents an intrinsic property of a genuine fingerprint obtained from finger imprint image. Let  $p(x, y)$  be a pixel in a fingerprint image. The local ridge orientation at  $p(x, y)$  is the angle  $\beta_{xy}$  that the fingerprint ridges crossing through small neighbourhood centered at  $p(x, y)$ , form with the horizontal axis [3]. The orientation field of the fingerprint image pixels has

been divided into directions  $z = (0, 1, \dots, 7)$  in a  $9 \times 9$  window, where the pixel  $p(x, y)$  in the center of the window if located [5]. Idea of the pixel orientation computing by means of mask is presented in Fig.1. Each element of the mask from Fig. 1 corresponding to the pixel  $p(x, y)$  located over fingerprint image. The orientations are restricted to eight numbered directions:  $d_z = z \times 22.5^\circ$ ,  $z = \{0, 1, 2, 3, 4, 5, 6, 7\}$ . Hence,  $z \rightarrow d_z$ .

In the first step, by means of the mask from Fig.1 the gray-level fingerprint image values in the eight directions (at positions marked by numbers 0,1,...,7) are determined. For example, for pixel  $p(x, y)$  from Fig.1 appropriate values have been computed:

$$\begin{aligned}
 s_0 &= p(x - 2, y) + p(x - 4, y) + p(x + 2, y) + p(x + 4, y) \\
 s_1 &= p(x - 2, y + 1) + p(x - 4, y + 2) + p(x + 2, y - 1) + p(x + 4, y - 2) \\
 &\quad \vdots \\
 &\quad \vdots \\
 s_7 &= p(x - 2, y + 1) + p(x - 4, y + 2) + p(x + 2, y - 1) + p(x + 4, y - 2)
 \end{aligned}$$

Finally, the values  $s_i$  are added according to index  $i$ . The sums  $S_i$  are equivalent to convolving the image with  $9 \times 9$  masks, where each mask has value of 1 at positions where  $9 \times 9$  window block has values  $i$  and 0 elsewhere.

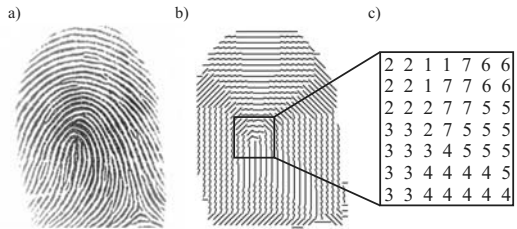
In the next stage, the min/max value of each  $i$  is estimated:

$$i = \begin{cases} s_d = \arg \max_{i=\{0,1,\dots,7\}} \{S_i\} & \text{for } p(x, y)^{bin} = 1 \\ s_l = \arg \min_{i=\{0,1,\dots,7\}} \{S_i\} & \text{for } p(x, y)^{bin} = 0 \end{cases} \quad (1)$$

where:  $p(x, y)^{bin}$  is a pixel value at point  $p(x, y)$  in fingerprint image.

The orientation at pixel is defined by means of  $s_d$  value if the central pixel is located on a ridge (black area), and by the  $s_l$  value if the central pixel is located in a valley. Unfortunately, obtained values, treated as pixels orientation, are usually noisy, therefore they should be smoothed and averaged in a local neighbourhood. In our application, the angles between the blocks  $9 \times 9$  are smoothed by passing the image through a mode function [5].

In the next stage background is eliminated and only foreground is used. If in a given  $9 \times 9$  block at least one black pixel can be found, then whole block creates part of orientation field, otherwise such block is treated as background and is rejected. After calculations, appropriate directions are stored in the table  $TG[x, y]$  as numbers  $z$ . The table of the orientation field can be presented as image (although it is not necessary, where each value is performed as appropriate line section. For example, if element  $TG[x, y] = 0$  - then horizontal line is performed, for element  $TG[x, y] = 4$  - vertical line is drew, and so on for remained directions. Above mentioned operations are presented by Fig. 2, where fingerprint image, orientation field and fragment of such field were shown.



**Fig. 2.** a) Image of fingerprint. b) Appropriate orientation field of analyzed fingerprint as image and its selected fragment. c) Fragment of the orientation field from 70:fig:b performed as table.

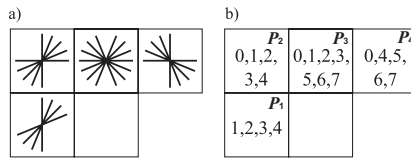
### 3 The Reference Point Determination by Means of the Neighbourhood Influence Filter

Reference point location is very important task in analysis of fingerprint image. Several fingerprint-matching algorithms align fingerprint images according to a landmark, called the core [3]. In practice, the core point corresponds to the center of the point of the north most loop type singularity. For fingerprints that do not contain loop or whorl singularities (e.g. belonging to the so-called arch type fingerprint class), it is difficult to define the core. For this reason the core

is usually associated with the point of maximum ridge line curvature [3, 5]. In this paper the new method of the fingerprint reference point location has been proposed. These methods are still investigated and developed [2, 4, 5, 6].

### 3.1 Filter Description

In the proposed method the filter  $Mr$  will be used. Additionally, this filter will be called as filter of neighbourhood influence. Elements of the filter  $Mr$  have been experimentally determined, and depend on direction of fingerprint ridges. These directions are appropriate selected and are different in different filter area. The graphic, structural characteristic of the filter is depicted in Fig.2a. Distribution of values in the filter elements presents Fig.2b.



**Fig. 3.** a) Graphic representation acceptable fingerprint ridge directions in the filter  $Mr$ . b) An adequate numeric values for the fingerprint ridges.

The appropriate elements  $P_x, x = \{1, 2, 3, 4\}$  of the filter  $Mr$  are the sets:

$$P_1 = \{1, 2, 3, 4\}, P_2 = \{0, 1, 2, 3, 4\},$$

$$P_3 = \{0, 1, 2, 3, 5, 6, 7\}, P_4 = \{1, 4, 5, 6, 7\}.$$

In practice, except the filter  $Mr$ , two additional auxiliary tables  $TR_i, i = 1, 2, \dots, 7$  and  $TA$  are needed. The values  $r_x^i, x = 1, 2, 3, 4$  are stored in tables  $TR_i$  were experimentally established. Fig.3b presents selected values for each table  $TR_i$ . Fig.3a presents adequate distribution of the fingerprint ridges.

The auxiliary table  $TA[x, y]$  has dimension of the table  $TG[x, y], x = 1, 2, \dots, X, y = 1, 2, \dots, Y$ . At the beginning, all elements of the table  $TA$  has values of 0 for image background and of 1 for fingerprint area.

The filter  $Mr$  is simultaneously moved in to the right inside of the areas of the tables  $TG$  and  $TA$ , beginning from the left upper corner and successive is moved down to the lower right corner of the tables. Current value of the element  $C$  of the auxiliary table  $TA$  is successive changed in calculation process. The influence of the neighbouring cells of the table  $TA$  on the cell  $C$  is located in this cell as value  $vC$ . If some element  $TG[x, y]$  has the same value as one of the set  $P_x$ , then the  $vC$  value in the cell  $C$  of the auxiliary table  $TA$  is determined on the basis of another cells of the table  $TA$  and the appropriate values  $r_x^i$  of the table  $TR_i$ .



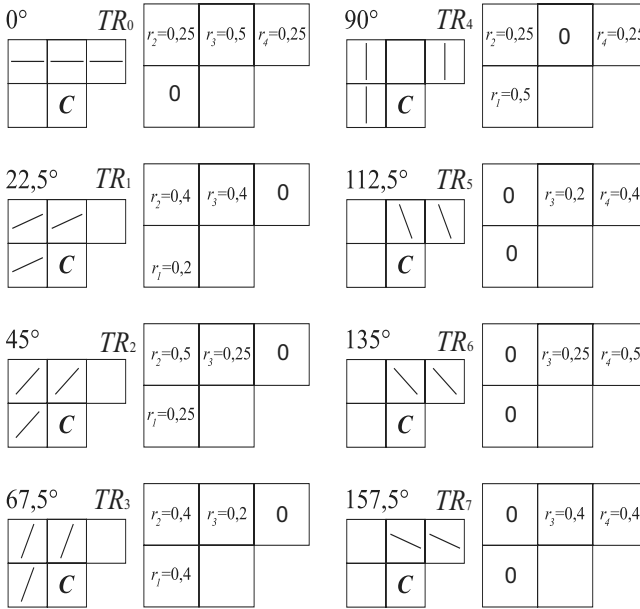


Fig. 4. Selection of the values  $r_x$  in the tables  $TR_i$

### 4 Calculations by Means of the Filter $Mr$

Let the elements  $z_1, z_2, z_3, z_4$  are elements of the table  $TG$ ,  $z_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ . For these conditions, the new value of the cell  $C$  is determined from equation:

if in the table  $TA$  condition:

$$a_i \neq 0 \tag{2}$$

for  $i = 1, 2, 3, 4$  is fulfilled, then:

$$vC^{new} = \sum_{i=1}^4 r_{z_i \in P_i}^i \times a_i \tag{3}$$

where  $a_i$  is the value of the appropriate cell of the table  $TA$  Fig.4b.

If above mentioned condition is not fulfilled, then value (2) is not calculated and filter  $Mr$  is shifted to the right, to next position. After shifting, the value (3) located in the cell  $C$  of the table  $TA$  can be computed again but condition (2) should be always fulfilled.

#### Example 1

Some fragments of the tables  $TG$  and  $TA$  presents Fig.4a and Fig.4b, respectively. The table  $TG$  includes numbers treated as numeric fingerprint ridge

directions. In this example the values of the table  $TG$  were randomly generated. For these conditions the new value  $vC^{new}$  of the cell  $C$  can be computed.

$$z_1 = 4 \rightarrow z_1 \in P_1, a_1 = 1.5$$

$$z_2 = 3 \rightarrow z_2 \in P_2, a_2 = 0.8$$

$$z_3 = 4 \rightarrow z_3 \notin P_3, a_3 = 0.5$$

$$z_4 = 7 \rightarrow z_4 \in P_4, a_4 = 2.0$$

$$vC^{prev} = 1$$

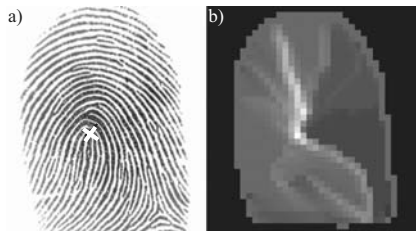
$$vC^{new} = \sum_{i=1}^4 r_{z_i \in P_i}^i \times a_i = (r_{4 \in P_1}^1 \times a_1) + (r_{3 \in P_2}^2 \times a_2) + (r_{4 \notin P_3}^3 \times a_3) + (r_{7 \in P_4}^4 \times a_4) = (0.5 \times 1.5) + (0.4 \times 0.8) + (0 \times 0.5) + (0.4 \times 2) = 1.87$$

a)						
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="padding: 5px;">3</td><td style="padding: 5px;">4</td><td style="padding: 5px;">7</td></tr> <tr><td style="padding: 5px;">4</td><td style="padding: 5px;">2</td><td style="padding: 5px;"></td></tr> </table>	3	4	7	4	2	
3	4	7				
4	2					

b)												
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="padding: 5px;"><math>a_2</math></td><td style="padding: 5px;"><math>a_3</math></td><td style="padding: 5px;"><math>a_4</math></td></tr> <tr><td style="padding: 5px;">0,8</td><td style="padding: 5px;">0,5</td><td style="padding: 5px;">2</td></tr> <tr><td style="padding: 5px;"><math>a_1</math></td><td style="padding: 5px;">C</td><td style="padding: 5px;"></td></tr> <tr><td style="padding: 5px;">1,5</td><td style="padding: 5px;">1</td><td style="padding: 5px;"></td></tr> </table>	$a_2$	$a_3$	$a_4$	0,8	0,5	2	$a_1$	C		1,5	1	
$a_2$	$a_3$	$a_4$										
0,8	0,5	2										
$a_1$	C											
1,5	1											

**Fig. 5.** a) fragment of the table  $TG$ . The values of the table cells result from fingerprint ridge directions, b) the current values of the auxiliary table  $TA$ .

After filter  $Mr$  relocation in whole area of the table  $TG$ , the auxiliary table  $TA$  will be populated. The largest value in the table  $TA[x, y]$  determined where fingerprint reference point is located. Place where fingerprint reference point has been found was indicated in Fig.6. In Fig.6a this place is marked as cross. Fig.6b presents the results of the neighbourhood influence method on reference point determination, where appropriate gray-scale values form the image. The brightest point in this image indicate also reference point of the fingerprint. The coordinates  $(x, y)$  of such point correspond to the same coordinates localised



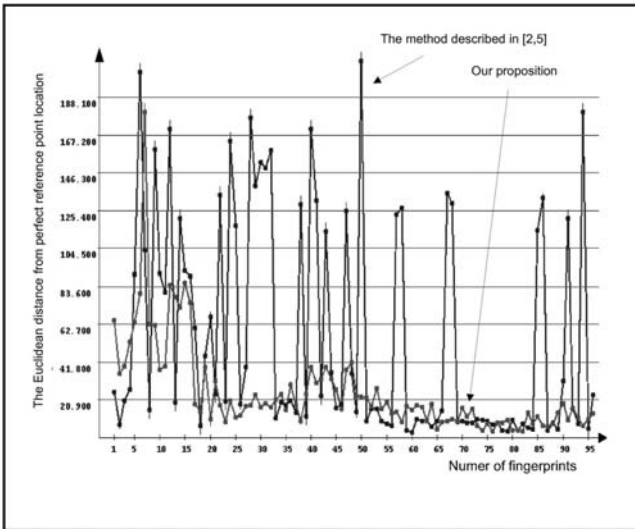
**Fig. 6.** a) Fragment of the table  $TG$ . The values of the table cells result from fingerprint ridge orientations. b) The values of the auxiliary table  $TA$ .

fingerprint reference point at the image Fig.6a. Hence, fingerprint reference point coordinates  $r_p(i, j)$  can be immediately obtained from the table  $TA$ :

$$\max_{i,j}(TA[i, j]) \rightarrow \{i, j\} \rightarrow r_p(i, j) \tag{4}$$

### 5 Conclusion and Investigation Results

Proposed method of the fingerprint reference point location was compared to the method described in [2, 5, 6, 7], where reference point was also searched. In mentioned works some filters called masks were additionally compared with method, which was prepared on the basis of [3]. Hence, described in this paper investigations compared these two methods. Because these methods indicate the same point, methods can be immediately compared. The reference point has pixel coordinates  $x, y$ . For this reason the A, B and E sets of the data have been gathered. The set A includes reference points indicated by our method, the second set (B) includes points collected by means of method [5]. Additionally, the set E of reference points is accessible, where the police expert’s coordinates are gathered. Finally, the Euclidean distance between appropriate coordinates can be formed:  $[(x_a - x_e)^2 + (y_a - y_e)^2]^{0.5}$  or  $[(x_b - x_e)^2 + (y_b - y_e)^2]^{0.5}$ . In the next stage, the sets A and B with the set E have been compared. From Fig.7 follows clearly that proposed method of reference point determining gives results closest located to the police expert assessment.



**Fig. 7.** Accuracy of the two methods of the reference point determination compared to police expert assessment

Since human experts may not be able to locate the pixel wise accurate reference point, we propose the new method which allow to determine such point more accuracy.

## References

1. Jain A.K., Prabhakar S., Hong L., Pankanti S. (2000) Filterbank-Based Fingerprint Matching. *IEEE Trans. on Image Processing*. Vol. 9, No. 5, pp.846-859.
2. Jiang X., Liu M., A.C., Kot A.C. (2004) Reference Point Detection for Fingerprint Recognition. *Proc. of the 17th Int. Conf. On Pattern Recognition (ICPR'04)* Cambridge, UK, pp. 540-543.
3. Maltoni D., Maio D., Jain A.K, Prabhakar S.(2003) *Handbook of Fingerprint recognition*. Springer-Verlag.USA
4. Park C.H., Oh S.K., Kwak D.M., Kim B.S., Song Y.C., Park K.H. (2003) A New Reference Point Detection Algorithm Based on Orientation Pattern Labeling in Fingerprint Images. *Proc. of 1st Iberian Conf. On Pattern Recognition and Image Analysis, Puerto de Andratx,Spain*, pp.697-703.
5. Porwik P., Wieclaw . (2004) A new approach to reference point location in fingerprint recognition. *IEICE International Journal Electronics Express*. Japan. Vol. 1, No. 18, pp. 575-581.
6. Porwik P.,Wrobel K. (2005) The new algorithm of fingerprint reference point location based on identification masks. *4th International Conference on Computer Recognition Systems CORES'05. Advances in soft Computing*. Springer-Verlag. Vol. XVII, pp. 807-814.
7. Porwik P. (2007) Combined fingerprints recognition method basing on image filtering, minutiae analysis and reference point location. *Elektronika. SIGMA-NOT*. No. 4. Warsaw. Poland, pp. 31-39.

---

# Gene Expression Programming-Based Method of Optimal Frequency Set Determination for Purpose of Analogue Circuits' Diagnosis

P. Jantos, D. Grzechca, T. Golonek, and J. Rutkowski

The authors are with the Faculty of Automatic Control, Electronics and Computer Science Silesian Technical University, Gliwice, Poland  
{pjantos, dgrzechca, tgolonek, jrutkowski}@polsl.pl

**Summary.** This paper presents a method for optimal frequency search with the use of Gene Expression Programming (GEP). It includes a brief summary of material of analogue circuits fault diagnosis and introduction to GEP. Developed method belongs to SBT diagnosis routines' group. Frequency responses for chosen hard faults of a circuit under test as well as for an intact circuit are simulated before test. There are ambiguity sets provided to avoid a fault tolerance masking effect.

## 1 Introduction

An ongoing miniaturization and an increasing complexity of analogue electronic circuits (AEC) cause a necessity of developing reliable and efficient diagnostic and testing methods. There are well known testing procedures of digital circuits and devices. However, methods of testing analogue and mixed circuits have not been developed, yet. There are several problems to solve in the aim of developing a standardised AEC diagnosis methods. The most important of them are a variety of analogue signals and elements, a larger set of possible faults in comparison to digital circuits, a limited accessibility to the measurement points, and a fault tolerance masking effect [1].

Generally, there are three aims of diagnosis a fault detection, a fault localization, and a fault identification. A fault detection is the most basic diagnosis test. The aim of it is to determine whether the circuit under test (CUT) is damaged or intact, the "GO/NO GO" test. The other question is to find a source of a fault. Answer to this problem is obtained with the fault localization procedure. Once the fault is localized it may be important to identify the fault (determining a type and a value of the fault). Taking under consideration used diagnosing tools, three groups of diagnosis procedures can be distinguished: simulation before test (SBT) methods, simulation after test (SAT) methods, and a built-in self-test (BIST). The SBT methods are based on simulations of a certain and previously chosen set of CUT faults before the explicit test phase. It allows for shortening the diagnosis total time. An aim of the SAT methods is in most cases

an identification of faults. These methods are efficient in soft faults identification. A disadvantage of these routines is high computing cost and long analyses' time. The methods using boundary path require designing a whole circuit in the way that allows for independent diagnosis of chosen test blocks (often test blocks are chosen in the way they are equal to functional blocks) [2].

A circuit testing may be performed in a time domain, a frequency domain (AC testing) and a direct currents domain (DC testing). The DC testing makes it possible to obtain results with high reliability level. The AC or time domain testing, though, allows for gathering more information about the CUT state than DC testing without a need of measurements in more than one test point.

The presented procedure of finding optimal frequency set for purpose of AEC testing is a dictionary (signature) method. The method may be implemented as a preparation for signature dictionary constructing in SBT routines. A very important asset of presented method is a self-adapting size of optimal frequencies' set.

## 2 Gene Expression Programming

The Gene Expression Programming (GEP) is an evolutionary algorithm (EA). GEP integrates features of genetic algorithm (GA) and genetic programming (GP). In following parts of this chapter a brief introduction to GEP is presented, implementation issues and a fitness function described [3].

### 2.1 Individuals Coding

Gene Expression Programming individuals are built with chromosomes (genes strings) of fixed length. The chromosomes are decoded to phenotypes of tree structure, i.e. expression trees (ET). There is not requirement of using all genes of chromosomes in the process of expressing individuals. Therefore, chromosomes of the same length code ET of different structure and complexity level.

Genes that chromosomes are coded with may be divided into two groups functions set  $\mathbf{H}$ , and a terminals set  $\mathbf{T}$ .

The functions set may contain any functions possible to describe over the elements that belong to the terminals set. It may be arithmetic operations, set operations, logic operations, etc. Both, variables and constants may be elements of  $\mathbf{T}$ .

There was a set of frequencies coded with each individual of a population in the presented method. It was decided that set  $\mathbf{H}$  will contain only one element, that is a sum of sets. With this operator it was possible to add a new frequency to the set  $\mathbf{F}'$  (an optimized frequencies set).

$$\mathbf{H} = \{\cup'\} \quad (1)$$

Where  $\cup$  is a sum of two sets.

Terminals set was described as a sum of set  $\mathbf{F}$  and an "empty" element which did not add any frequency to  $\mathbf{F}'$ .

$$\mathbf{T} = \mathbf{F} \cup \{\emptyset'\} \tag{2}$$

Where  $\emptyset$  is an empty terminal that does not code any frequency.

There is a head and tail of a chromosome distinguishable in the body of chromosomes. There are no limits that would determine the length of the head. It is important, though, that the head is started with an element of the set  $\mathbf{H}$ . The rest of the head elements may belong to either  $\mathbf{H}$  or  $\mathbf{T}$  sets. The tail of chromosome may be built only with elements of the set of terminals. Its length is given with equation:

$$\|tail\| = \|head\| \cdot (n - 1) + 1 \tag{3}$$

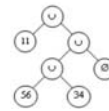
Where:  $\|tail\|$  is a length of the tail;  $\|head\|$  is a length of the head;  $n$  - is a maximum number of arguments of the used functions set elements.

	0	1	2	3	4	5	6	7	8	9	0	1	2	
Head	∪	11	∪	∪	∅	56	34	8	7	23	97	87	40	Tail

**Fig. 1.** An exemplary GEP chromosome

there is an exemplary GEP chromosome presented.

There is also an encoded expression tree presented in the Fig. 2. The set  $\mathbf{F}'$  described with the chromosome has 3 elements (frequencies):  $\{11, 34, 56\}$ . In our research all individuals have been built with 4 chromosomes. The chromosomes head length was decided to be equal to 7. Therefore, one individual allows for coding up to 32 different frequencies. The population size equals 40 individuals.



**Fig. 2.** A phenotype - ET - of the exemplary chromosome from the Fig. 1

### 2.2 Genetic Operations

References [3] and [4] suggest there is no difference between different selection methods. It is strongly advised to use a simple elitism in any GEP implementation. The elitism means copying the best (or few best) individual to the offspring population without modifying them. There are a few genetic operators used in GEP a mutation, crossovers, transpositions.

There is an uniform mutation implemented usually. It is important, though, to keep a functions set element on the first chromosome position. In the presented implementation mutation possibilities were as listed: function to frequency (and vice versa), function to empty terminal (as above), and frequency to empty terminal (as above).

A crossover in GEP is not different from a crossover operator in regular genetic algorithm. The crossover does not add new genetic material to the gene pool of

population. There are 3 different types of crossover implemented a single point crossover, a two point crossover, and a chromosomes crossover.

A transposition operator causes copying the part of chromosome of certain length to the other position of the same chromosome. The transposition operators with mutation add a new genetic quality to the population’s genetic pool. Transpositions that are used in GEP are an insertion sequence (IS) transposition, a root insertion sequence (RIS) transposition, and a chromosomes transposition [3],[4].

### 2.3 Fitness Function

A fitness function (FF) is the most important part of any EA application. As it was previously mentioned a FF choice determines a proper AE work. In the presented approach the task for FF was to find such a  $\mathbf{F}'$  set that the healthy circuit is separated (obligatory condition), find such a  $\mathbf{F}'$  set that the most hard faults are localized and a distance between each two of them maximized, and find an  $\mathbf{F}'$  of least possible power (additional condition).To cover above points the FF has been designed:

$$F_{fitness} = 20(U + 1) + 10P \tag{4}$$

Where:  $U$  - a number of correctly localized CUT states;  $P$  - a penalty modifier.

$$P = A [a (1 - e^{-\Delta_{min}}) + b (1 - e^{-\Delta_{avg}})] \quad \wedge \quad a + b = 1 \tag{5}$$

$$\Delta_{min} = \min_i (\Delta m_i \quad \wedge \quad f_i \in \mathbf{F}') \tag{6}$$

$$\Delta_{avg} = \frac{\sum_{i=1}^K (\Delta m_i \quad \wedge \quad f_i \in \mathbf{F}')}{\|\mathbf{F}'\|} \tag{7}$$

Where  $a, b$  are coefficients (in our approach  $a=1$  and  $b=0$ ).

$$A = \begin{cases} Z & \text{if } S_0 \text{ localized} \\ -0.5 & \text{if } S_0 \text{ not localized} \end{cases} \tag{8}$$

In case of not localizing  $S_0$  (the non-faulty circuit) the value of  $Y$  ( $Y=-0.5$ ) has been chosen heuristically. The value of  $X$  is computed basing on a number of frequencies coded with being assessed individual. The value was chosen heuristically.

$$\begin{cases} Z : 1.00 & 0.70 & 0.65 & 0.50 & 0.30 & 0.10 \\ D : D \leq 2 & D = 3 & D = 4 & 5 \leq D \leq 7 & 8 \leq D \leq 12 & 13 \leq D \end{cases} \tag{9}$$

Where  $D$  is a number of  $\mathbf{F}'$ .

Fitness function given with above equations allows for fulfilling all of the set conditions. Is the faulty-free CUT localized properly the number of  $\mathbf{F}'$  is



decreasing and the distances between localized faults maximized. If the intact CUT is not localized the number of  $\mathbf{F}'$  is increasing and the distances between localized faults minimized. Therefore, with the more frequencies  $S_0$  state is more likely to be separated correctly [3],[4].

### 3 Optimal Excitation Choice Strategy

The procedure of choosing testing signal frequencies may be divided into two stages. Both of them are presented in the following paragraphs.

#### 3.1 Before Test Stage

At this stage a new diagnostic system is created. The system should allow for obtaining maximum information about the CUT with the use of a minimal set of frequencies. It was assumed that the CUT has  $N$  test points  $M_i$ . Amplitude responses have been measured at all accessible nodes  $N$ . The set of test points is represented by a vector  $\mathbf{M}$ :

$$\mathbf{M} = \{M_1, M_2, \dots, M_N\} \tag{10}$$

The most common strategy is measuring a characteristic on the CUT output ( $N=1$ ). Hence, CUT states set is given by a vector:

$$S = \{S_0, S_1, \dots, S_L\} \tag{11}$$

where  $S_0$  represents an intact circuit state. The latter states ( $S_1, \dots, S_L$ ) represent hard faults. Short elements has been simulated by the parallel conductance of 1000 [S] and open elements with serial resistance of 1 [TΩ]. The amplitude characteristic have been measured in chosen test points  $M_i$ . The single step between two following frequencies has been set to 1 [Hz] (frequency shift) over the range  $f_{min}$  and  $f_{max}$ :

$$f_{min} = 10^X \quad \wedge \quad f_{max} = 10^Y \quad \wedge \quad X, Y \in C \quad \wedge \quad X < Y \tag{12}$$

A total points number of amplitude characteristic is as follows:

$$K = \frac{f_{max} - f_{min}}{\Delta f} \quad \wedge \quad \Delta f = 1 [Hz] \tag{13}$$

Hence, frequencies set may be given with a vector:

$$\mathbf{F} = \{f_1, f_2, \dots, f_K; \wedge \|f_{i+1} - f_i\| = \Delta f \wedge f_1 = f_{min} \wedge f_K = f_{max}\} \tag{14}$$

For the exemplary circuit considered in section 4  $X=0$  and  $Y=4$  (Eq. 12). So,  $f_{min}=1$  [Hz] and  $f_{max}=10$  [kHz]. The total number of frequencies is  $K=10000$ . A task and an aim of the method is to choose such a subset  $\mathbf{F}' \in \mathbf{F}$  that a fault separability and detectability remain at the same levels as with the use of full set  $\mathbf{F}$ .

To determine an influence of elements tolerance on CUT amplitude characteristic a number of Monte Carlo analyses (MCA) has been computed. Tolerance value of resistive elements is  $tol_R$  and reactive elements (capacitors and coils)  $tol_X$ . For the circuit considered in chapter 4  $tol_R = 2\%$  and  $tol_X = 5\%$ . A result of MCA simulation were envelopes, top and bottom, of amplitude CUT answers on parameters' values changes. A difference of above values is determining an ambiguity areas set. This set for each of  $K$  frequencies may be given with equation:

$$\Delta M = \{\Delta m_1, \Delta m_2, \dots, \Delta m_K\} \tag{15}$$

Two states are separable if for a frequency  $f_i$  the distance (amplitude response) between them is greater than  $m_i$ . Dictionary is created based on optimal frequency selection.

### 3.2 After Test Stage

A quality of proposed method is valuated. A series of tests is provided. An analysis of their results may lead to answers: whether the method is correct, what a test construction computation cost is, and what possibilities of implementation of the algorithm are. An assessment of found solution bases on a fitness function (presented in following chapter) that takes into account the previously calculated margin as an ambiguity regions set [5].

## 4 Exemplary Circuit Diagnosis

The presented method has been tested with the use of an exemplary circuit (Fig. 3) [1]. Chosen discrete elements have been taken into account (see Tab. 1 for details). Based on 16 elements 33 states of CUT are created (healthy circuit and short and open faults of each element). The number of set F was  $K=10000$  frequencies with the coefficients  $X=0$  and  $Y=4$ . Hence,  $f_{min} = 1$  [Hz],  $f_{max} = 10$  [kHz]. A global solution (with the use of all possible frequencies) allowed for localization of 26 hard faults. The rest of faults were grouped into 3 ambiguity sets. There are amplitude characteristics of all CUT states presented in the Fig. 4.A. The ambiguity region over the predefined range of frequencies (the distance between envelopes) is shown in the Fig. 4.B.

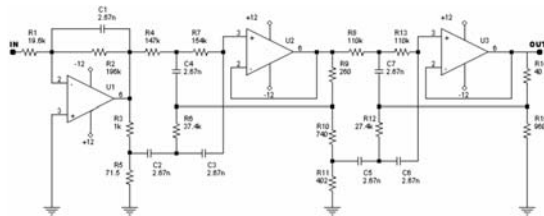
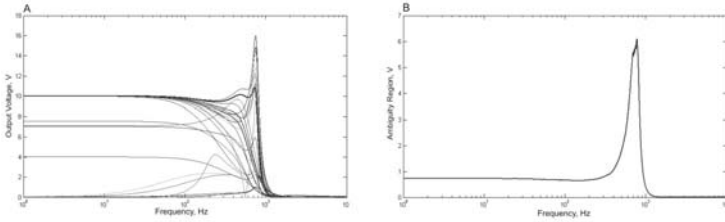


Fig. 3. The diagnosed exemplary circuit [1]

**Table 1.** The selected test circuits' elements<sup>1</sup>

Element R1, R2, . . . , R8, R12, R13, C2, . . . , C7

<sup>1</sup> Odd states represent short elements, even state -open ones. State  $S_0$  represents faulty-free circuit.



**Fig. 4.** The intact and damaged CUT amplitude characteristics (A); the ambiguity region (B)

**Table 2.** The results of CUT (Fig. 4.1) testing with the use of all possible frequencies

Correctly localized CUT states	$S_0, S_1, S_4, S_5, S_7, S_{10}, \dots, S_{14}, S_{16}, \dots, S_{19}, S_{21}, \dots, S_{32}$
Found ambiguity sets	$\{S_2, S_3\}, \{S_6, S_9\}, \{S_8, S_{15}, S_{20}\}$

**Table 3.** The results of GEP based method

No.	Used frequencies [Hz]	Number of localized CUT states	Time <sup>2</sup>
1	93.0;518.0;1528.0;3492.0	26	2.76
...	...	...	...
10	106.0;350.0;1530.0;3271.0	26	2.61

<sup>2</sup> Time [sec] of 100 generation proceeding.

Tab. 2. presents diagnosis' results if all frequencies have been used. The method presented in the previous section should minimize the number of frequencies while the diagnosis' results must remain on the same level.

There have been run 10 independent processes for finding minimal  $F'$  set with a limit of 1000 generations set. The time of 100 generation proceeding was measured. In the Tab. 3 there are chosen results of such works submitted. Found frequencies vector  $F'$  consist of 4 elements allow for localizing 26 hard faults in each case. The results are the same as with the use of all possible frequencies. The average 100 generation processing time has been 2.67 second.

It is important that GEP based method allowed for finding the minimal  $F'$  sets composed of small number of elements. Moreover, GEP allows on automatic changing the number of  $F'$  set without changing the length of individuals (i.e. chromosomes). A computation cost of the presented method is low. Hence, it should not influence total diagnosis costs.

## 5 Conclusions

The new method for the finding an optimal frequency set for purpose of analogue circuits' fault diagnosis has been presented. The GEP is used as a search engine. The aim of the paper was not to compare the procedure with existing ones. The proposed method was tested with the use of the exemplary circuit. The results may lead to a conclusion that employed algorithm may be used to create an efficient tool for optimal frequency set finding. The algorithm assets are: an ease of implementation, a short proceeding time (which may suggest that there is a sense of developing a GEP based industrial applications), and a self-adapting optimal frequencies' set.

## References

1. Catelani M., Fort A.: Soft Fault Detection and Isolation in Analog Circuits: Some Results and a Comparison Between a Fuzzy Approach and Radial Basis Function Networks, IEEE Trans. On Instrumentation and Measurement, vol.51, no. 2, pp. 196-202, 2002
2. Kabisatpathy, P., Barua, A., Sinha, S., Fault diagnosis of analogue integrated circuits, Springer 2005, ISBN: 038725742X
3. Ferreira, C., Gene Expression Programming: Mathematical Modelling by an Artificial Intelligence, ISBN: 9729589054
4. Ferreira, C. Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. Complex Systems 2001, Vol. 13, Issue 2, pp. 87-129
5. Golonek, T., Grzechca, D., Rutkowski, J., Evolutionary Method for Test Frequencies Selection Based on Entropy Index and Ambiguity Sets. ICSES 2006, International Conference on Signals and Electronic System, d, Poland 2006, pp. 511-514

---

# Prediction of Economic Situation of Small and Medium Enterprises Using Bayesian Network

Barbara Kuczmowska<sup>1</sup>, Andrzej Burda<sup>2</sup>, and Zdzisław S. Hippe<sup>3</sup>

<sup>1</sup> College of Management and Public Administration in Zamość, Akademicka 4, 22-400 Zamość, Poland

[bkuczmowska@wszia.edu.pl](mailto:bkuczmowska@wszia.edu.pl)

<sup>2</sup> College of Management and Public Administration in Zamość, Akademicka 4, 22-400 Zamość, Poland

[aburda@wszia.edu.pl](mailto:aburda@wszia.edu.pl)

<sup>3</sup> University of Information Technology and Management in Rzeszów, Sucharskiego 2, 35-225 Rzeszów, Poland

[zhippe@wsiz.rzeszow.pl](mailto:zhippe@wsiz.rzeszow.pl)

**Summary.** The structure of a belief network model for predicting the economic situation of small and medium enterprises in the Lubelskie and Podkarpackie provinces is discussed. The direction of further scientific research in order to improve performance of a belief network model is additionally presented.

## 1 Introduction

Various analytical and knowledge acquisition methods were used to develop models for assessment of the enterprise's economic and financial situation. The Bayesian networks, further referred to as BN, are one of them. The research on BN applications to predict the situation of enterprises were conducted among others by Sarkar [4] and Sun [2].

## 2 Scope and Objective of the Research

This research is aimed at development of a Bayesian belief network model for predicting the financial situation of small enterprises from Lubelskie and Podkarpackie provinces. For the needs of research two states of enterprise were specified: "good situation" and "poor situation". Enterprises of poor economic and financial condition, in which at the same time the values of the three parameters: of the financial gross result, the sale exchange rate, and the employment rate changes are negative. The enterprises of good financial condition are such, in which at least one of these parameters is positive. Economic and financial conditions are described  $Y$  using two-state variable, which assumes the value:

- 0, if the enterprise is in a poor economic and financial condition,
- 1, if the enterprise is in a good economic and financial condition.

The statistics data used to build the model come from the Regional Statistics Centers in Lublin and Rzeszów. The research covered small and medium-sized enterprises up to (from 10 to 49 persons) in two provinces Lubelskie and Podkarpackie, in the years of 1999-2004. Due to the fact that usually the character of the applied the descriptive attributes is represented by relative values related to the previous year, to the research were chosen only enterprises, that submitted financial statements, and were active in the subsequent years starting from 1999 year.

The research was conducted on a sample of 380 cases - enterprises registered under the heading no. 51 of the Business Register “Wholesale trade and consign-ment services, excluding trade in cars and motorcycles”. For their description 13 attributes were chosen to present various aspects of its activity and the mezzoeconomic situation in the year  $t-1$  (Table 1), and decision variable  $Y$  describing the financial and economic condition of the company in the year  $t$ .

**Table 1.** Descriptive attributes in the Bayesian models

Type of parameter	Symbol	Name of parameter
Financial liquidity	wsk12	Current ratio
Parameters describing enterprise’s property	wsk2	Share of inventory in total assets
	wsk18	Productivity of total assets
Parameters describing the capital structure	wsk8	Net worth to fixed assets
	wsk10	Share of working capital in total assets
	wsk11	Share of working capital in current assets
Parameters describing effectiveness of resources	wsk19	Labor productivity
	wsk20	Value added intellectual coefficient
	wsk21	Productivity of fixed assets
Parameters of profitability	wsk24	Gross profit margin
	wsk26	Return on total assets
Macroeconomic variables	z6	Rate of export
	z7	Average rate of exchange (\$)

### 3 Description of the Experiment

The research was conducted using the *BeliefSEEKER* system [3] applying of the 10-fold cross validation method.



**Table 2.** Classification statistics all networks for the testing subset

Model	Class	Quantity		% share	
		Correct	Incorrect	Correct	Incorrect
1.	Poor	12	6	60.00	30.00
	Good	15	4	83.33	22.22
2.	Poor	12	8	57.14	38.10
	Good	13	4	76.47	23.53
3.	Poor	13	8	61.90	38.10
	Good	10	7	58.82	41.18
4.	Poor	13	9	59.09	40.91
	Good	14	2	87.50	12.50
5.	Poor	18	1	81.82	4.55
	Good	10	6	62.50	37.50
6.	Poor	8	4	57.14	28.57
	Good	21	2	87.50	8.33
7.	Poor	13	4	76.47	23.53
	Good	16	4	76.19	19.05
8.	Poor	13	5	68.42	26.32
	Good	16	2	84.21	10.53
9.	Poor	10	6	62.50	37.50
	Good	21	1	95.45	4.55
10.	Poor	17	2	85.00	10.00
	Good	15	3	83.33	16.67
Avg.	Poor	13.00	5.00	66.95	27.76
	Good	15.10	3.50	79.53	19.60

**Table 3.** Classification statistics of selected networks for the data set

Network	Class	Quantity		% share	
		Correct	Incorrect	Correct	Incorrect
S.10 – Single network	Poor	134	52	70.53	27.37
	Good	160	29	84.21	15.26
Avg. for 10 networks	Poor	131.30	56.20	69.11	29.58
	Good	152.60	35.90	80.32	18.89
Voting method	Poor	129	61	67.89	32.11
	Good	166	24	83.27	12.63

direct impact on the classification result. The classification statistics for these networks are presented in the Table 2.

The Table 3 presents the classification results for the best belief network, selected in the first phase of the research and an average result for 10 networks calculated for the full data set.



## 5 Discussion

The analysis of the belief network with the Dirichlet parameter  $\alpha = 60$  leads to a conclusion that two parameters have direct influence on the financial and economic situation of an enterprise: ‘Return on total assets’ (wsk26) and ‘Gross profit margin’ (wsk24). An indirect impact on the enterprise situation can be revealed in case of the following parameters: ‘Net worth to fixed assets’ (wsk8), ‘Value added intellectual coefficient’ (wsk20) and ‘Share of inventory in total assets’ (wsk2). In some networks additional impact have attributes: ‘Share of working capital in total assets’ (wsk10) and ‘Share of working capital in current assets’ (wsk11). On the other hand, ‘Current ratio’ (wsk12), ‘Labor productivity’ (wsk19), ‘Productivity of fixed assets’ (wsk21) and ‘Average rate of exchange (\$)’ (z7) have no influence on the decision node.

The average value of an error for all networks was 19.60% in case of Type I error and 27.76% in case of Type II error (Table 2). The value of Type II error is considerably higher than Type I error. The network no. 10 shows distinguishing results: the classification error for “poor situation” is 10% and for “good situation” is 16.67%. This network was selected as the best one.

The Table 3 shows the test result for the best network (full data set) and the average result for all 10 belief networks. The classification error of a single network for the full data set increased significantly for “poor situation” cases (from 10% to 27.37%) and improved slightly for “good situation” cases. Similar trends were observed for the average error values, although the differences are lower. The average Type I error decreased from 19.60% to 18.89% and the average Type II error increased from 27.76% to 29.58%.

In the chart (Fig.2) the results have been summarized obtained with the voting method compared with the self-sufficient network (S.10). In case of Type I error the voting method shows advantage over the self-independent network model. But in case of Type II error this method did not bring the expected effects.

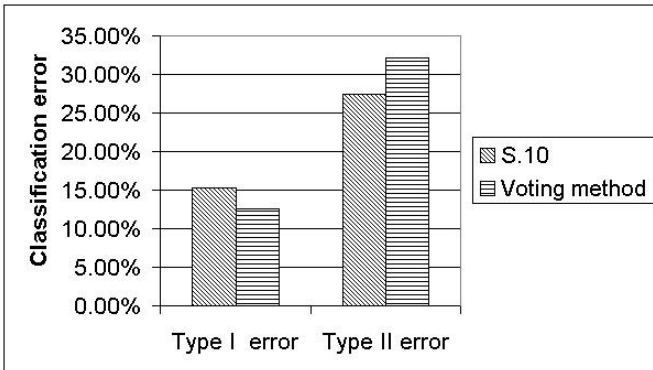


Fig. 2. Classification error for Bayesian models

The voting method requires modification. The weights related to the classification error in individual networks (ranging from 4.55% to 41.18%) should be taken into account.

All training models and their combinations show considerably worse performance in case of poor financial situation. The approach to discretization of continuous attributes used in model development. Earlier research showed that discretization has a significant influence on the models generated.

Further research on belief networks will focus on the improvement of models' ability to classify "poor situation" cases correctly through a change of the approach to discretization of the primary data and modification of a set of attributes describing the economic situation of an enterprise. In addition to that a work will be undertaken to develop a methodology of the Bayesian network ensemble using the voting method.

**Acknowledgement.** This research was financed under the **EQUAL** project No. F0086 implemented by the **College of Public Administration and Management in Zamość**.

## References

1. Heckerman D. (1995) A tutorial on Learning Bayesian Networks. Technical report MSR-R-95-06.
2. Sun L., Shenoy P.P. (2004) Using Bayesian Networks for Bankruptcy Prediction. School of Business, Working Paper No 295, University of Kansas.
3. Hippe Z.S., Grzymała-Busse J.W., Mroczek T. (2006) Belief Networks vs. Relief Rules: Basic Theory and Practical Issues. In: Zuba D., Parczewski A. (Eds.) Chemometrics. Methods and Applications. Institute of Forensic Research Publishers, Kraków, pp.47-67.
4. Sarkar S., Sriram R.S. (2001) Management Science 47:1457-1475.

---

# Ensembles of Artificial Neural Networks for Predicting Economic Situation of Small and Medium Enterprises

Andrzej Burda<sup>1</sup>, Barbara Kuczmowska<sup>2</sup>, and Zdzisław S. Hippe<sup>3</sup>

<sup>1</sup> College of Management and Public Administration in Zamość, Akademicka 4, 22-400 Zamość, Poland  
aburda@wszia.edu.pl

<sup>2</sup> College of Management and Public Administration in Zamość, Akademicka 4, 22-400 Zamość, Poland  
bkuczmowska@wszia.edu.pl

<sup>3</sup> University of Information Technology and Management in Rzeszów, Sucharskiego 2, 35-225 Rzeszów, Poland  
zhippe@wsiz.rzeszow.pl

**Summary.** Results of predicting an economic and financial situation of small and medium size enterprises from Lubelskie and Podkarpackie regions are presented. As opposed to other research in this field, the micro-macro (mezzo-) modeling concept was used. The results of prediction using neural perceptrons and two types of ensembles of artificial neural networks were compared. Some conclusions and directions of further research aiming at improving quality and extending the functionalism of enterprises being investigated are dealt with.

## 1 Introduction – Research Scope and Objectives

Initially, models for assignment the economic situation of small and medium-sized enterprises developed with the use of various analytical methods, focused on the search for one only parameter being the basis for this assessment [4]. However, since the assessment of the same enterprise but based on of different parameters can lead to contradicting results, attempts were made to apply a multidimensional analysis. A particular attention should be paid to the following models: logit [7], discrimination with a k-Nearest-Neighbor method [9], decision-tree classification [5], Bayesian belief network [8] or artificial neural networks [2]. Comparative analyses of these models frequently provide ambiguous conclusions, depending on the prediction horizon, type of enterprises or their location [3].

However, available literature on the subject provides many examples of high effectiveness of neural models for predicting enterprise bankruptcy. For this reason, a research project was dealt with to obtain reliable and effective prediction models, based on multilayer perceptron architecture, for predicting the economic and financial situation of the enterprises in the Polish economy. The analysis

covered two types of artificial neural network ensembles. Their performance was then compared to those generated by the best single artificial neural network developed in the course of the experiment.

Most of the researchers in this field attempt to predict the bankruptcy of enterprises. Usually, the court's decision announcing the initiation of the bankruptcy proceedings or the bankruptcy itself are accepted as the bankruptcy criteria. Such approach has a good foundations but is not free of important disadvantages. This was *inter alia* the reason why the concept of prediction of enterprise's economic and financial situation was a matter of our interest.

The term of "economic and financial situation of an enterprise" is not defined unambiguously in the literature yet. For the needs of this research we follow the approach that distinguishes between two states: "good" or "poor economic and financial situation" [6]. The analyses presented in the paper mentioned above suggest that an enterprise in a poor situation should be defined as an enterprise, which reports three parameters to be negative, namely, the gross profit, the rate of turnover and the rate of employment. On the other hand, if at least one of these parameters is positive, it is assumed that the economic and financial situation of the enterprise is not jeopardized.

The research covered 1866 cases of small and medium-sized enterprises employing from 10 to 49 employees, registered under the heading no. 51 of the Business Register "Wholesale trade and consignment services, excluding trade in cars and motorcycles" in the region of Lubelskie and Podkarpackie.

## 2 Description of the Experiment

### 2.1 Description of the Informational Database

The concept of the economic and financial situation adopted in the previous chapter allows to describe it as a binomial variable, which may take on the following values:

- 0, if an enterprise is in a poor economic and financial situation,
- 1, if an enterprise is in a good economic and financial situation.

The decision table was built on the basis of the data reported in financial statements of the enterprises. A set of economic and financial attributes was calculated for each enterprise.

It should also be taken into account that in each year the enterprises operate in different environment defined by the general macroeconomic situation of the country, as well as the mezzo-economic specifics of the regional economy and the social situation. This is why the set of variables describing the individual situation of an enterprise include a number of macro- and mezzo-economic variables, identical for all enterprises in the region. Such approach, based on micro-macro (mezzo-) modeling, enjoys increasingly higher interest of socio-economic researchers.

All cases in the decision table are described by a vector of numbers, which components are real values representing various aspects of the enterprise's business

operations and the macro- and mezzo-economic situation in the year  $t-1$ . The last component of the vector is equal to 0 or 1 and represents the economic and financial situation of the enterprise in the year  $t$ . This approach to data collection enabled the development of a model with one-year prediction horizon.

## 2.2 Selection of Cases

Among 1866 cases obtained from the Regional Statistics Centres for the period of 1999-2004 only 190 (88 in the Lubelskie region and 102 in Podkarpackie) turned out to be in poor economic and financial situation, as defined in the Section 1. To ensure a well-balanced sample a subset was created comprising all these cases. Then, the “good situation” cases were randomly selected and added to this subset (proportionally to the total number of cases in both regions). In this way the primary data set **C\_380\_33** was built and comprised 380 cases described with 33 explanatory variables and one dependent variable (binomial), namely, the economic and financial situation of an enterprise.

## 2.3 Selection of Attributes

Among the various methods for input space minimization in a neural model, analyzed in our previous research, the best results were achieved in case of the model sensitivity analysis. To ensure better performance of the process the selection of variables was divided into two phases. The first phase aimed at preliminary evaluation of the usability of descriptive variables. For this purpose the correlation between all 33 attributes and the dependent variable  $Y$  was determined.

In real sets of financial and economic data some data points may deviate significantly from the trend. Since the presence of such data points may affect the analysis of Pearson correlation, the Spearman’s rank correlation was used. This method provides good results even in the presence of a large number of outliers, which is not the case for the Pearson’s correlation.

In this phase all variables, which were not correlated significantly with the dependent variable  $Y$  or for which the correlation coefficient did not exceed 0.15, were excluded.

The descriptions of selected attributes and the corresponding Spearman’s rank correlation coefficients are shown in the Table 1. As a result of this analysis a reduced data set (**C\_380\_13**) was created including 13 input variables (11 microeconomic and 2 macroeconomic indicators) and one output variable  $Y$ .

The second phase was implemented in the course of the network training process by means of a network sensitivity analysis [1].

## 2.4 Description of the Research

The Statsoft’s **Statistica 7.1** software was used in the research. The process of training a 3-layer perceptron (two hidden layers, 8 neurons each) was conducted for the **C\_380\_13** data set using a 10-fold cross validation method in a supervised mode.

**Table 1.** Descriptive attributes in the neural model

Type of parameter	Symbol	Name of parameter	Correlation with Y
Financial liquidity	wsk12	Current ratio	0.2267
Parameters describing enterprise's property	wsk2	Share of inventory in total assets	0.1501
	wsk18	Productivity of total assets	0.2648
Parameters describing the capital structure	wsk8	Net worth of fixed assets	0.2555
	wsk10	Share of working capital in total assets	0.1685
	wsk11	Share of working capital in current assets	0.2145
Parameters describing effectiveness of resources	wsk19	Labor productivity	0.2440
	wsk20	Value added intellectual coefficient	0.5419
	wsk21	Productivity of fixed assets	0.2414
Parameters of profitability	wsk24	Gross profit margin	0.5295
	wsk26	Return on total assets	0.5608
Macroeconomic variables	z6	Rate of export	0.1649
	z7	Average rate of exchange (\$)	0.1937

Ten independently classifying networks, trained by means of the cross validation method (NN<sub>1</sub> ÷ NN<sub>10</sub>), were configured into two network ensembles, thus creating so-called combined models: the Ensemble I (ENN<sub>I</sub>), which output value was generated through “voting” and the Ensemble II (ENN<sub>II</sub>), which output was an average value of output neuron activation in all component networks. This value, assuming the normal distribution of the input variables, may be interpreted as the probability of that the case belongs to a particular class. With two classification thresholds set up (0.49 as the upper limit of the input signal for the enterprises in a poor economic and financial situation and 0.56 as the lower limit for the enterprises in a good economic and financial situation), the Ensemble II cannot recognize objects in the range between these two thresholds. Analogously, the Ensemble I does not recognize objects at voting equilibrium.

For each network and for both ensembles, the necessary classifying statistics were calculated and two types of error were distinguished: Type I and Type II errors. It is worth mentioning here that Type II error (wrong classification of “poor situation” enterprises) is more dangerous, which is due to its costs.

The classification error for the sample was adopted as the criterion for selection of the best-performing independent network. In case of comparable error values (within 1%) the network with the lowest error of second type was selected.

**2.5 Test Results**

The network sensitivity analysis resulted in reducing the number of inputs by from eight to ten variables. The Table 2 shows the influence of particular parameters on the quality of the model.

**Table 2.** The influence of particular parameters on the quality of the model

	NN	wsk2	wsk10	wsk11	wsk12	wsk18	wsk19	wsk21	wsk26	z6	z7
1.	1.01	1.00	1.01	1.02	1.00	1.01	1.00	1.10	1.00	1.01	
2.	1.18	1.13	1.11	1.07	1.08	1.14		1.51	1.29	1.30	
3.	1.16	1.09	1.09		1.20	1.13		1.45	1.25	1.10	
4.	1.18	1.08	1.13		1.20	1.16		1.53	1.25	1.18	
5.	1.38	1.33	1.36		1.44	1.46		1.69	1.52	1.58	
6.	1.12	1.07	1.08		1.11	1.13		1.45	1.31	1.36	
7.	1.35	1.24	1.23		1.18	1.21		1.65	1.47	1.44	
8.	1.15	1.13	1.11		1.13	1.14		1.41	1.34	1.26	
9.	1.19	1.13	1.16		1.09	1.06		1.67	1.36	1.08	
10.	1.31	1.21	1.23		1.29	1.32		1.56	1.45	1.51	

An image of network classification created by means of the 10-fold cross validation method shows differential values. The Type I error was in the region 6% to 44% and the Type II error from 11% to 39%.

The Table 3 includes classification statistics for two types of network ensembles described in the Subsection 2.4 and for the best single network, chosen on the basis of the criterion adopted in this chapter. To enable a comparative analysis of the models a modified NN\_8 model was introduced (NN.8a), in which one classification threshold was replaced with two thresholds set at 0.44 and 0.51. A percentage share of non-recognized cases is a complement of the sum of shares presented in the Table 3 to 100%, while the quantity of such shares a complement of the sum of quantities to 190.

**3 Discussion**

The neural models developed in the research, as assumed in Subsection 2.1, are designed to predict the economic and financial situation of selected enterprises within the prediction horizon of 1 year.

The analysis of data confirms the thesis that combined models show much better prediction quality than their single component networks. The average error value for the component networks was 24.46% (Type I error) and 27.53% (Type II error). These errors are much lower for network ensembles - 12% and about 7%, respectively. It is worth mentioning that the Type II error, which leads to especially dangerous consequences, dropped significantly. The comparisons of

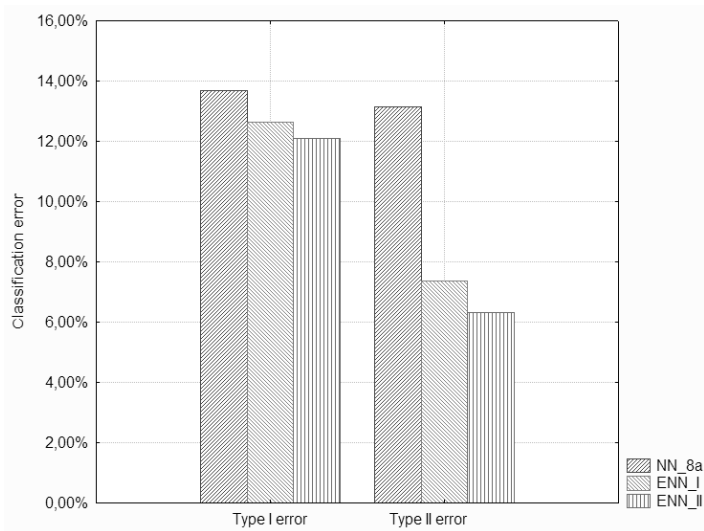
**Table 3.** Classification statistics for the network ensembles for **C\_380\_13**

Model	Class	Quantity		% share	
		Correct	Incorrect	Correct	Incorrect
NN_8	Poor	164	26	86.32	13.68
	Good	160	30	84.21	15.79
NN_8a	Poor	159	25	83.68	13.16
	Good	158	26	83.16	13.68
ENN_I	Poor	162	14	85.26	7.37
	Good	159	24	83.68	12.63
ENN_II	Poor	174	12	91.58	6.32
	Good	154	23	81.05	12.11

network ensembles (Table 3) to the best-performing single network NN\_8a are depicted on the Fig.1. They also prove that the network ensembles have clear advantage over the single perceptron models, particularly in relation to the error of second type.

The prediction quality of both types of neural network ensembles fluctuates within similar boundaries. A slight difference lies in the fact that the Ensemble II returned significantly higher percentage (91.58%) of correct classifications while the level of errors was similar. It resulted from a huge reduction of non-recognized cases.

The analysis of incorrectly recognized cases for all models showed that 66% of them were assessed similarly by all models. It may suggest that these



**Fig. 1.** Classification errors for neural models



cases are strongly noised or biased with the errors already described in the literature.

The coefficients representing the influence of individual attributes on the network quality (sensitivity quotients), although subjected to a high variance, show that practically all networks based on the cross-validation method used the same eight input parameters. It should be noted that the dominant role is played by the ‘Return on total assets’ (wsk26) and two macroeconomic variables: ‘Rate of export’ (z6) and the ‘Average rate of exchange (\$)’ (z7). On the other hand, the parameters showing high correlation to the dependant value (such as: ‘Value added intellectual coefficient’ and ‘Gross profit margin’) were excluded in the course of the analysis of network sensitivity. All *mezzo*-economic variables were rejected due to a low Spearman’s rank correlation coefficient with the dependent value Y.

It should be mentioned here that the available literature on the generation of neural learning models for predicting the bankruptcy of Polish enterprises does not provide examples of a utilization of *mezzo*- and macro-variables.

## 4 Conclusions

The analysis of the results presented in the previous section confirms the usability of neural perceptrons for predicting the economic and financial situation of the enterprises registered under the heading no. 51 of the Business Register “Wholesale trade and consignment services, excluding trade in cars and motorcycles” in the region of Lubelskie and Podkarpackie. The advantage of the network ensembles over the single neural networks encourages the use of such models. As the effectiveness of both types of ensembles is similar, it is recommended to use the Ensemble II type, a because they provide additional information on the probability that the enterprises belong to a particular class of good or poor enterprises.

The level of prediction errors may suggest that some significant factors affecting the prediction were not taken into account. A crucial element of the further research shall be a deeper analysis of a full set of factors, including their combinations and, possibly, new ones.

Therefore, further research on the neural models for predicting the economic and financial situation of the enterprises will be continued through extending it with ensembles of other architectures and learning modes, such as: radial base function networks (RBF), probabilistic neural networks (PNN) and linear networks. Further improvement of the prediction quality may be achieved through continued work on optimization of a heterogeneous combined model utilizing various methods of benchmark classification.

**Acknowledgement.** This research was financed under the **EQUAL** project No. F0086 implemented by the **College of Public Administration and Management in Zamość**.

## References

1. Lula P., Tadeusiewicz R. (Eds.) (2005) *Statistica Neural Networks<sup>TM</sup>*. StatSoft Poland, Kraków, pp.140-142 (in Polish).
2. Odom M., Sharda R. (1990) A neural network model for bankruptcy prediction. In: *International Joint Conference on Neural Networks, San Diego, Part II*:163–168.
3. Altman E.I., Marco G., Varetto F. (1994) *Journal of Banking & Finance* 18:505–529.
4. Beaver W.H. (1996) *Supplement to Journal of Accounting Research* 4:71–111.
5. McKee T., Greenstein M. (2000) *Journal of Forecasting* 19:219–230.
6. Nowak E. (2006) *Barometr Regionalny* 6:35–41 (in Polish).
7. Ohlson J.A. (1980) *Journal of Accounting Research* 18:109–131.
8. Sarkar S., Sriram R.S. (2001) *Management Science* 47:1457–1475.
9. Tam K.Y., Kiang M. (1990) *Applied Artificial Intelligence* 4:265–282.

---

# The Performance of the Haar Cascade Classifiers Applied to the Face and Eyes Detection

Adam Schmidt and Andrzej Kasiński

Institute of Control and Information Engineering str. Piotrowo 3a 60-965 Poznan,  
Poland

Schmidt.Adam@gmail.com, Andrzej.Kasinski@put.poznan.pl

**Summary.** Recently we have presented the hierarchical face and eye detection system based on Haar Cascade Classifiers. In this paper we focus on the optimization of detectors training. Moreover, we compare the performance of Lienhart's face detectors [1] and Castrillon-Santana's eyes detectors [2] with those which have been trained by us.

## 1 Introduction

The serial way of information processing in such face recognition systems results in cumulating errors of consecutive processing steps. Campadelli et al. [3] discuss the influence of the eyes localization error on face recognition (FR) ratio of some methods. He also concludes that some published FR results do not clearly report the fact of manual initialization performed, which results in better than average recognition rates reported by particular authors.

Anyway, in order to develop an automatic face recognition system, one has to design an efficient face and eye localization method.

In [4] we have presented hierarchical face and eyes detection system based on Viola's Haar Cascade Classifiers (HCC) [5] augmented by some knowledge-based rules.

Our main goal was to train the reliable face and eyes detecting HCCs and to use them in the above mentioned hierarchical system. Therefore, two different strategies for creating the training sets were tested. Moreover, the influence of weak classifiers complexity and the desired detection ratio of particular stage classifier on the overall performance of the detection system were assessed.

As neighboring positive responses of HCC are merged into a single detection, the influence of constraining minimum number of neighbors on the systems efficiency was investigated.

In this paper we present results of our tests. Firstly, we describe a new database of 10000 face images with manually marked face and eyes positions for reference. Secondly, we compare the results of face detection by using original Lienhart's detectors [1] with those which were trained on our image base. Then we compare the results of eye detection using Castrilon-Santana's HCC [2]

with our custom eye HCCs. The trials have been performed for eyes detection on both manually selected and automatically detected ROIs of the faces.

## 2 The Environment

In order to get statistically significant results, our classification method performance evaluation was conducted on a set of face images consisting of almost 10000 images of 100 people . The images were acquired in partially controlled illumination conditions, over uniform background, and stored as 2048x1536 pixels JPEG files. Each person's pictures were taken in the following sequences, while:

- turning their head from the right to the left,
- nodding their head from the raised to the lowered position,
- turning their raised head from the right to the left,
- turning their lowered head from the right to the left,
- moving their head without any constraint on the face pose.

The main purpose of creating such an image base was to provide an extensive and credible data for the systematic performance evaluation of the face detection, facial features extraction and face recognition algorithms. The number of gathered pictures guarantees sufficient inter- and between-class variability to obtain statistically reliable performance estimates of tested algorithms. In order to provide ground truth for the face detection and eyes extraction tasks the rectangle ROIs containing face and eyes were manually marked on each image in the base . The coordinates and dimensions of the rectangles circumscribing face and eyes were saved in OpenCV Storage files in the YAML format.

We implemented our system in the Visual C++ 6.0 by using the OpenCV [6] with Lienhart's implementation of the HCC. All detectors were trained using tools included in the OpenCV.

## 3 Detectors Training

The training a HCC involves the application of large and diverse sets of positive and negative samples (images ). With our new image base it was easy to get faces and eyes examples. However, creating a set of "non-faces" is a tricky task, because what does it mean to represent every possible non-face object? As all images in our base were taken with the same background, we wanted to check whether the negative set built from the same images with hidden faces is sufficient enough to distinguish between faces and non-faces. Another negative training set was created by randomly gathering about 3500 diverse pictures not containing any faces. Negative example sets for eyes detector were created by hiding left or right eye on the available face image. The following face detectors have been trained:

- C1 - hidden faces negative set, 20x25 window, level min hit rate 0.995 and stump as a weak classifier
- C2 - hidden faces negative set, 20x25 window, level min hit rate 0.995 and 2-split CART as a weak classifier
- C3 - hidden faces negative set, 20x25 window, level min hit rate 0.995 and 4-split CART as a weak classifier
- C4 - rich negative set, 20x25 window, level min hit rate 0.990 and 4-split CART as a weak classifier
- C5 - rich negative set, 20x25 window, level min hit rate 0.995 and 4-split CART as a weak classifier
- C6 - rich negative set, 20x25 window, level min hit rate 0.999 and 4-split CART as a weak classifier

Only single parameter at a time was changed during the experiments. The variant giving best results was used in the subsequent tests. The C1, C2 and C3 detectors differ in the complexity of weak classifiers used. The C3 was trained with the hidden faces negative set, while the C4 was trained using the rich negative set. The C4, C5 and C6 HCCs vary in the required TP ratio of a single stage classifiers.

Their performance was compared to the following Lienhart's detectors already available in the OpenCV:

- L1 - stump based, 24x24 window, Discrete AdaBoost
- L2 - stump based, 20x20 window, Gentle AdaBoost
- L3 - 2-split CART based, 20x20 window, Gentle AdaBoost
- L4 - 2-split CART based, 20x20 window, Gentle AdaBoost, with a tree made of stage classifiers instead of a cascade

Independently, the following eyes detectors were trained and the results compared with Castrillon-Santana's detector results (S):

- E1 - level min hit rate 0.995, with 4-split CART as a weak classifier
- E2 - level min hit rate 0.999, with 4-split CART as a weak classifier

Each of our detectors was trained with the Gentle AdaBoost to the theoretical FP ratio of  $10e-6$ .

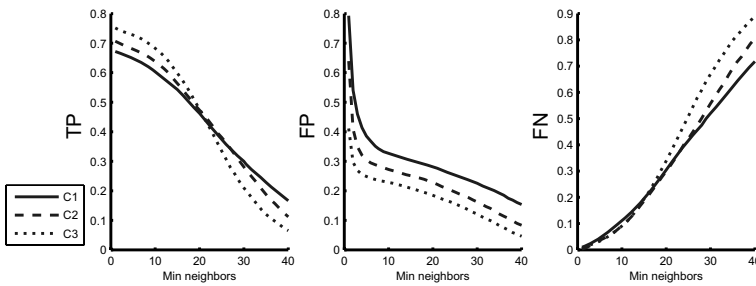
## 4 Results

### 4.1 Face Detection Results

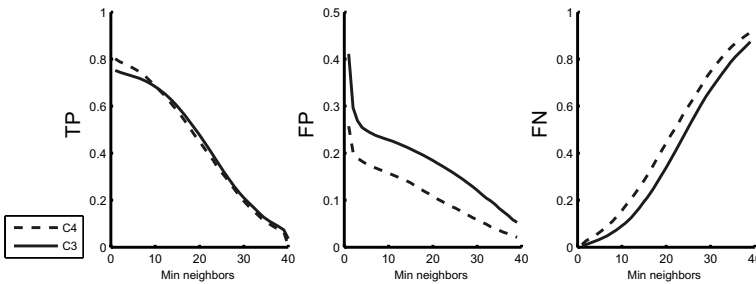
We applied Lienhart's and our detectors to the whole image base. Minimal detection window's size was 400x500 pixels for our detectors and 400x400 for Lienhart's. As Lienhart's detectors were trained on square windows, their outputs aspect ratio was reduced to 4:5 for compatibility. If intersection area of both the detected and ground truth rectangles was greater than 80% of both rectangles' areas a TP was claimed, otherwise the case was considered to be a FP. If no face was found on a whole picture, it was declared as false negative (FN).

**Table 1.** Average time of face detection on a PC with Intel Celeron 2800 MHz processor and 512 MB RAM

Detector	Average detection time [ms]
C1	246.07
C2	230.09
C3	227.49
C4	227.26
C5	214.15
C6	221.29
L1	503.93
L2	281.57
L3	237.87
L4	492.72



**Fig. 1.** The influence of weak classifier’s complexity on face detection ratios



**Fig. 2.** The influence of the training set diversity on face detection ratios

Influence of a minimum number of the neighbors on the face detection ratio was tested. The average time of the image processing with each detector was measured.

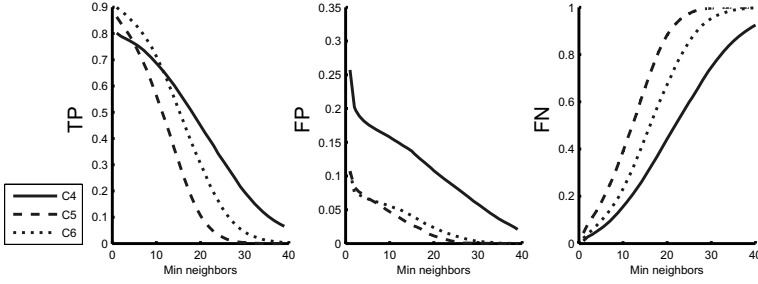


Fig. 3. The influence of the desired stage classifier’s TP ratio on face detection ratios

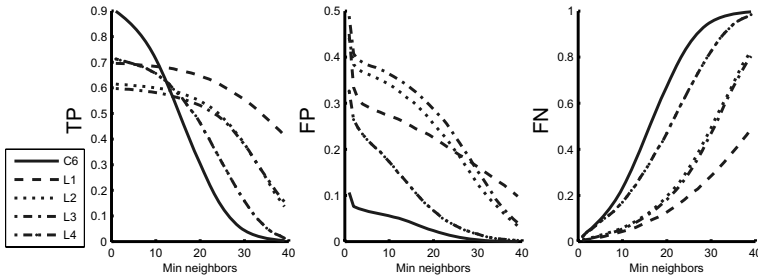


Fig. 4. The comparison of Lienhart’s face detectors with our best HCC

### 4.2 Eyes Detection Results

We compared the results of the Castrillon-Santana’s detectors with ours on the manually marked faces and the faces detected with C3, C6, L1 and L4 detectors. Error metric used here was the same as that of Campadelli[3]:

$$error = \frac{\max(\|C_l - C_{lGT}\|, \|C_r - C_{rGT}\|)}{\|C_{lGT} - C_{rGT}\|} \tag{1}$$

Where  $C_l$  stands for the center of the left eye found,  $C_r$  stands for the center of the right eye found,  $C_{lGT}$  and  $C_{rGT}$  for the centers of ground truth eyes.

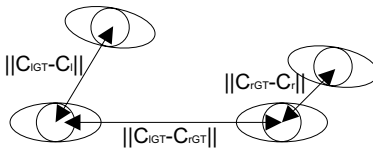
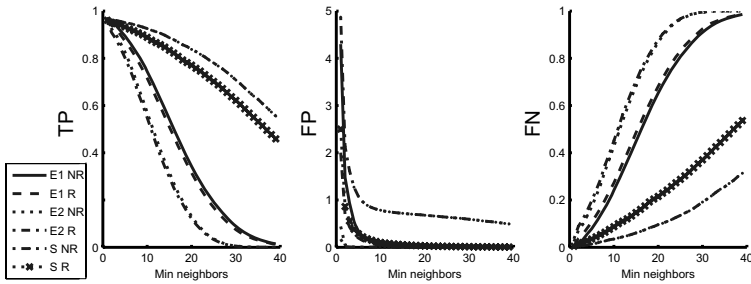


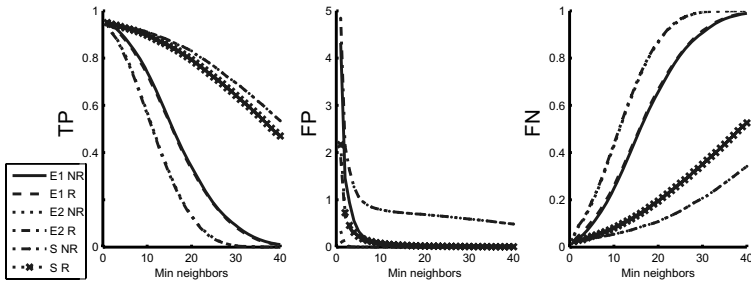
Fig. 5. Distances used in calculating the eye detection error

**Table 2.** Average detection time for eyes on a PC with Intel Celeron 2800 MHz processor and 512 MB RAM

Detector	Average detection time [ms]
E1 NR	377.35
E1 R	105.83
E2 NR	337.59
E2 R	100.06
S NR	2171.08
S R	625.07



**Fig. 6.** Eyes detectors performance while face ROIs have been manually marked beforehand

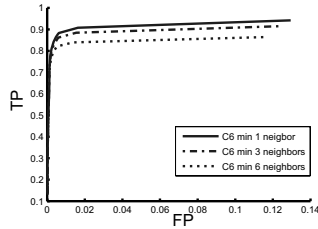


**Fig. 7.** Eyes detectors performance while face ROIs have been determined with the C6 face detector

Detections with relative error below 0.1 were treated as TPs, the others were considered as FsP. Pictures without any positive eyes detection were counted as false negatives (FNs).

Subsequently, the influence of minimum neighbors constraints of both face and eyes detectors on the eyes detection rate was tested. The average processing time of eyes detectors in both regionalized (R) and non-regionalized (NR) search was measured.





**Fig. 8.** TP ratio vs FP ratio of E2 R detector while face ROIs have been determined with different minimum neighbors constraint set

## 5 Conclusions

Our tests proved that using more complex weak classifiers pays off resulting in much more reliable detections. By increasing the number of splits in CARTs used as weak classifiers, higher TP ratio and lower FP ratio w.r.t the face detection were obtained. Many of FPs could be removed by restricting the minimum number of merged detections. However, it should be pointed out that only increasing it to the number of 5 produced positive results. Further increase of required neighbors resulted in the quick deterioration of the TP ratio without any significant change in the FP ratio.

Detectors trained on the plain background images turned out to be less efficient than those trained on rich negative set. This suggests, that in order to build a reliable classifier it is recommended to use as diversified negative set as possible.

By using 4-split CART as a weak classifier and setting the required TP ratio of each stage for 0.999 we were able to train the detector which outperformed all Lienhart’s detectors both w.r.t. the detection ratio and computational efficiency. By using solely the face detector we were able to detect 90% of the faces, getting the FP ratio of 11%.

Our results confirmed the hypothesis that using the regionalized search results in a significant reduction of the FP ratio and the processing time. Castrillon-Santana’s and our detectors reached comparable TP ratio, however our proposal turned out to have a several times lower FP ratio. It is worth to admit that processing time with our detectors was also 6 times shorter.

The eyes’ detection ratios, while restricted only to the face ROIs, were comparable despite various face detectors involved. It’s worth noting, that the best results were achieved with no restrictions on the minimum neighbors number of the face detector. This can be explained by the hierarchical nature of our system. Reducing the selectiveness at the first level (face detection) results in a propagation of dubious candidates. Thus more regions have to be evaluated at the second stage (eyes detection), which subsequently would discard face candidates with no eyes found. As the overall detection ratios of such a cascade are multiplicative, the TP ratio stays close to 100%, while the number of FP is reduced greatly. By using the combination of both our face and our regionalized

eyes detector we were able to fully automatically detect eyes in 94% of images still keeping the FP ratio of 13%. By applying minimum neighbors constraint solely to the eyes detector the TP ratio of 88% was achieved with less than 1% FP.

## References

1. Lienhart R, Kuranov A, Pisarevsky V (2002) Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. Technical report, Microprocessor Research Lab, Intel Labs
2. Castrillón-Santana M, Lorenzo-Navarro J, Déniz-Suárez O and Falcón-Martel A (2005) Multiple face detection at different resolutions for perceptual user interfaces. In: 2nd ICPRIA. Estoril, Portugal
3. Campadelli P, Lanzarotti R, Lipori G (2006 (preprint)) Eye localization: a survey. In: The fundamentals of verbal and non verbal communication and the biometrical issues. NATO Science Series
4. Kasinski A, Schmidt A (2007) The Architecture of the Face and Eyes Detection System Based on Cascade Classifiers. Submitted to CORES 2007
5. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of CVPR 1:511–518
6. The Open Computer Vision Library. [sourceforge.net/projects/opencvlibrary/](http://sourceforge.net/projects/opencvlibrary/)

---

# FPGA Implementation of the Gradient Adaptive Lattice Filter Structure for Feature Extraction

Boguslaw Szlachetko and Andrzej Lewandowski

Wroclaw University of Technology, Institute of Telecommunication,  
Teleinformatics and Acoustics  
{Boguslaw.Szlachetko, Andrzej.Lewandowski}@pwr.wroc.pl

**Summary.** In this paper evaluation issues of an FPGA implementation of the feature extraction gradient adaptive lattice (GAL) filter are presented. Two different hardware architectures for implementation in the Virtex family of FPGA devices were proposed, namely a single GAL section coprocessor of Virtex PowerPC, and a pipeline architecture. Both of them show significantly higher performance comparing to their software implementation.

## 1 Introduction

The application of digital signal processing algorithms to real world problem often requires implementing them in embedded systems. Current technological progress in the area of microcontollers, general-purpose processors, DSP processors and programmable logic device allows more and more complex DSP algorithms to be implemented in real-time systems.

One the classes of algorithms commonly used in modern signal processing are adaptive lattice filters. These filters can be applied to various applications: e.g. for speech enhancement (noise and echo cancellation) in speech recognition system [1, 2], as echo cancellers in passive airport surveillance radar with automatic target recognition or for feature extraction of speech signals in speech recognition systems [3]. In this latter approach the speech signal is the input to the adaptive lattice filter which estimates parameters (reflection coefficients) of the input signal.

In this paper we deal with an FPGA implementation of one type of adaptive lattice algorithms, namely gradient adaptive lattice (GAL) for feature extraction in the embedded recognition system.

## 2 AR Signal Model

In our work we assume that the input signals can be modeled as an autoregressive process:

$$\sum_{m=0}^M a_m(n)x(n) + \sigma(n) = 0 \quad (1)$$

where  $M$  is the model order,  $n$  denote time instant,  $a_m(n)$  are time varying autoregressive parameters,  $\sigma(n)$  is a white noise and  $x(n)$  is the signal of interest. Given the set of  $a_m(n)$  parameters, one can obtain LPC coefficients in other representations (autocorrelation sequence, reflection coefficients, poles of the transfer function) [4, 5]. In the rest of this paper term “feature” or “parameter” denotes the reflection coefficient.

This model is commonly used, for instance, in processing of digital communication signals, speech signals or EEG recordings. The model order  $M$  depends on a signal to be modeled. For speech signals the size of the model  $M$  is usually assumed to be in range  $< 8..12 >$  [6]. In the case of the analytic speech signal (with original input signal as a real part and its Hilbert transform as an imaginary part) it will be sufficient to estimate only half  $M = < 4.6 >$  of the number of coefficients. [5, 7]. For communication signals the model order depends on the type of modulation.

### 3 Gradient Adaptive Lattice Algorithm

The gradient adaptive lattice (GAL) algorithm is a type of stochastic gradient algorithm used for estimating reflection coefficients of the lattice predictor shown in Fig.1.

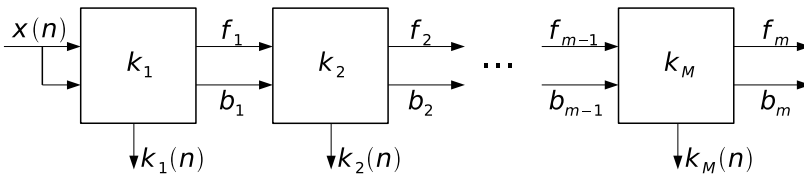


Fig. 1. Lattice filter structure for feature extraction

The internal architecture of the stage (section)  $m$  of the lattice predictor is presented in Fig.2.

Outputs signal of the section  $m$  can be expressed as follows

$$\begin{aligned}
 f_m(n) &= f_{m-1}(n) + k_m^*(n)b_{m-1}(n-1) \\
 b_m(n) &= b_{m-1}(n-1) + k_m(n)f_{m-1}(n)
 \end{aligned}
 \tag{2}$$

where  $f_m(n)$  and  $b_m(n)$  denote forward and backward prediction errors at the output of stage  $m$ ,  $f_{m-1}(n)$  and  $b_{m-1}(n)$  are forward and backward prediction errors at the input of the stage,  $b_{m-1}(n-1)$  is the delayed input backward prediction error, and  $k_m(n)$  is the reflection coefficient.

The adaptation rule of reflection coefficients is given by

$$k_m(n) = k_m(n-1) - \mu [f_{m-1}(n)b_m^*(n) + b_{m-1}(n-1)f_m^*(n)]
 \tag{3}$$

This rule is depicted in Fig.2 by the gray dashed lines.

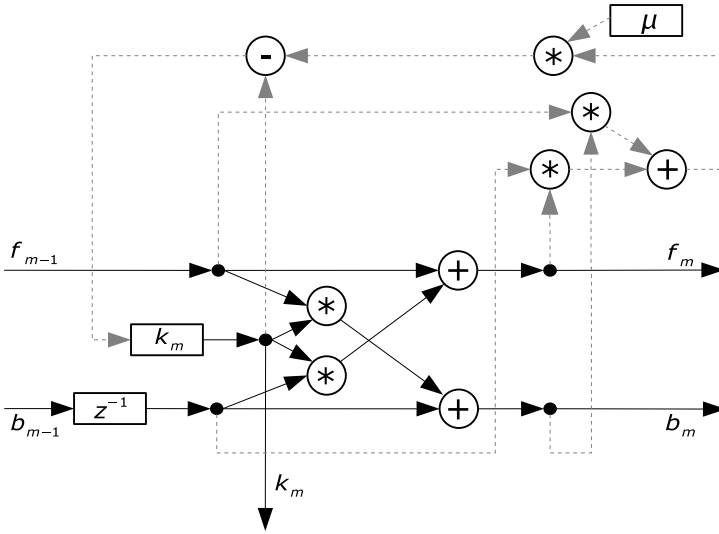


Fig. 2. Internal architecture of section  $m$

### 4 GAL Algorithm in an FPGA

Because of its structure, the GAL algorithm is well suited for pipelined implementation in an FPGA. In our research we chose the Virtex-II Pro family from Xilinx as the implementation platform. The main reason for such choice were additional hardware resources of Virtex-II Pro devices i.e. embedded PowerPC blocks and up to 444 hardware 18 bit x 18 bit multipliers, which allows efficient implementation of DSP embedded systems.

Prior to the FPGA implementation we implemented and tested the GAL algorithm in 16-bit fixed-point arithmetic in C++. This program was our point of reference for verification of correctness and the operation speed of the FPGA GAL filter. The benchmarks of this reference program are presented in Tab. 1.

Table 1. Computation time/frequency of one filter section

	$t$ [ns]	$f_{max}$ [MHz]
Pentium Celeron 2.40GHz	0.963793	1.037
AMD Athlon64 3500+	0.516528	1.936

In order to evaluate resources needed by a single section of the GAL filter and its performance we implemented the complex-valued, fixed-point GAL section in VHDL. The real and imaginary parts of the input and output signals are 16-bit numbers in IIQ15 notation (bit 15 is a sign bit, then there is a dot and a

fractional part), the real and imaginary parts of the reflection coefficients are stored as 32-bit numbers in I1Q31 notation.

The synthesis results of the filter section for Virtex-II Pro XC2VP100 device are presented in Tab. 2.

**Table 2.** Synthesis results of one filter section synthesised on Virtex-II Pro XC2VP100, with speed grade -6

Minimum period	25.076 ns (Maximum Frequency: 39.879 MHz)
Number of Slices	436 out of 44096 0%
Number of Slice Flip Flops	316 out of 88192 0%
Number of 4 input LUTs	760 out of 88192 0%
Number of bonded IOBs	355 out of 1040 34%
Number of MULT18X18s	26 out of 444 5%
Number of GCLKs	2 out of 16 12%

The maximum clock frequency for the filter section is about 40 MHz. It means that our FPGA GAL section is about 20 to 40 times faster than its reference C++ implementation.

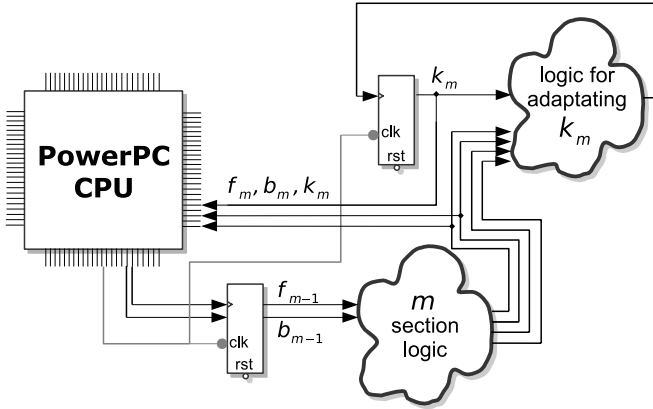
Following the above synthesis results, showing the allocated FPGA resources, one can observe that the FPGA utilization of logic resources is quite low. This means that one can consider employment of the remaining resources for an additional processing. The main issue is the utilization of the hardware multiplier blocks of the device - a single section of the GAL filter requires 26 multipliers. It means that for example XC2VP100, the biggest device of Virtex-II Pro family can contain up to 20 sections of the GAL filter. For more demanding real-world applications it could be possible to overcome this limitation by the cascade connection of a number of FPGA devices. In that approach, the Virtex-II Pro embedded RocketIO MultiGigabit Transceivers (which may operate at baud rates from 622 Mb/s to 3.125 Gb/s per channel) could be used.

The synthesis results obtained for a single section of the GAL filter allow two different hardware architectures of the GAL filter to be proposed. In the first one a single section acts as a coprocessor of the PowerPC embedded core. The second proposed architecture is the fully pipeline GAL filter.

#### 4.1 Single GAL Section Coprocessor

In the first approach we considered using a single section of the GAL filter algorithm as a coprocessor for the embedded PowerPC CPU (available in the devices of the Virtex-II Pro family). The input and output samples can be transferred to and from the processor via the Processor Local Bus. The block diagram of this solution is presented in Fig.3.

PowerPC program initializes section  $m$  with two input complex values  $f_{m-1}$ ,  $b_{m-1}$  according to the dataflow presented in Fig.2. Then the section is executed



**Fig. 3.** Section of the GAL feature extraction filter used as a coprocessor

and the CPU receives the section’s output values  $f_m$ ,  $b_m$  and the reflection coefficient  $k_m$ . The maximum section clock frequency is about 40 MHz. Assuming 300 MHz clock frequency of the PowerPC, processing time of the single filter section is about 8 PowerPC clock cycles. This means that at the same time the PowerPC CPU can be used to perform initialization data and storing the results in RAM.

The GAL filter section as a coprocessor would have to be examined more carefully to verify its properties, nevertheless, it seems that it has the following main advantages:

- low complexity (simplicity of handling the problem, allowing for further VHDL code optimization),
- flexibility - the filter order can easily be modified, as the filter length is controlled by a program executed on the PowerPC, and this value can be changed at any time-instant during the filter operation
- low resource usage of the FPGA device

The main limitation of this approach is that it can’t be pipelined. It means that for instance the GAL filter of order  $M$  can operate at maximum frequency  $f_{clk} = 40/M$  MHz.

### 4.2 Pipeline Architecture of GAL

In order to overcome the limitation of the solution presented in previous section we implemented fully pipeline architecture presented in Fig. 4. This architecture utilizes the fact that outputs signals of section  $m$  depend only on the section reflection coefficient and output signals of section  $m - 1$  as shown in (2) and (3).

This architecture works as follows: At the time instant  $n = 0$  the first sample  $x(0)$  arrives at the input of the section  $m = 1$ , then the section computes values  $f_1(0), b_1(0), k_1(0)$ . At the next time instant  $n = 1$  (after one clock period) section

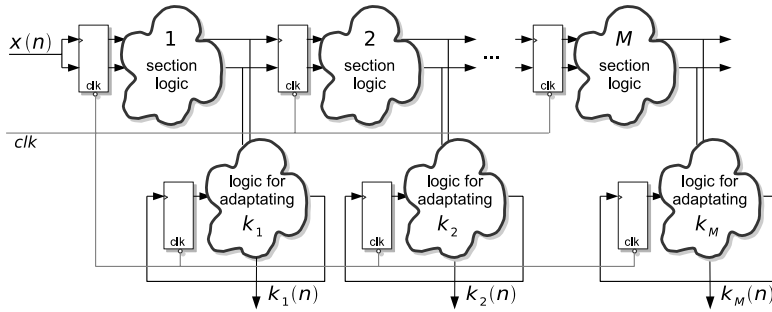


Fig. 4. Pipelined structure of the feature extraction lattice filter

$m = 2$  uses  $f_1(0), b_1(0)$  for computing values  $f_2(0), b_2(0), k_2(0)$  but at the same time the section  $m = 1$  can compute the  $f_1(1), b_1(1), k_1(1)$  and so on. Hence the first feature vector  $K(n) = [k_1(n), k_2(n), \dots, k_M(n)]$  will be available after  $M$  clock periods. After this initial startup time the pipeline is filled up so the feature vector  $K(n)$  for  $n > M$  will be updated at each clock period. The result is that, the  $M$ -sections filter data processing speed is equal to the processing speed of a single filter section.

The main advantage of the pipelined GAL architecture over solution presented in Sec. 4.1 is the fact that features vector  $K(n)$  can be estimated  $M$  times faster. Hence the maximum clock frequency for this architecture is  $f_{clk} \approx 40$  MHz on the Xilinx XC2VP100 device. The main drawback here is necessity of allocation of much more FPGA resources. This follows from the fact that, for the pipeline algorithm, all its sections have to be physically implemented in the FPGA structure. For example up to 20 GAL sections can be placed in Xilinx XC2VP100 but for XC2VP30 device the maximum filter order  $M = 4$ .

### 5 Conclusions

In this paper we presented FPGA implementation issues of the gradient adaptive lattice filter in the context of applying it for feature extraction in embedded recognition systems. We proposed two hardware architectures of an FPGA implementation of the GAL feature extractor. One of them consist in implementing a single GAL section as a coprocessor of the embedded PowerPC processor block. Such solution offers low usage of FPGA resources but the operation speed is a function of the filter order: GAL of order  $M$  can operate at maximum frequency  $f_{clk} \approx 40/M$  MHz. In the second approach we implemented the fully-pipeline architecture of the GAL filter. In this case the filter can operate at full speed  $f_{clk} \approx 40$  MHz at the costs of usage of FPGA resources. It should be noted that the operation speed of both proposed architectures outperforms the speed of the software implementation of the same GAL algorithm running on PC computers



with clocks greater than 2 GHz. Obtained results show that proposed architectures perform computations about 20 to 40 times faster than their PC software counterpart.

## References

1. A. Álvarez, V. Nieto, P. Gómez, and R. Martínez, "Speech enhancement based on linear prediction error signals and spectral subtraction," in *Proceedings of the International Workshop on Acoustic Echo and Noise Control*, pp. 123–126, 2003.
2. J. Hai and E. M. Joo, "Improved linear predictive coding method for speech recognition," in *Proceedings of Information Communications and Signal Processing, 2003*.
3. F. Runstein and F. Violaro, "An isolated-word speech recognition system using neural networks," in *Proceedings of the 38th Midwest Symposium on Circuits and Systems*, vol. 1, pp. 550–553, 1995.
4. C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*. Prentice Hall, 1992.
5. B. Szlachetko, *Parametric joint time-frequency transforms of stochastic signals*. PhD thesis, Wrocław University of Technology, 2001.
6. J. Zarzycki, *Orthogonal digital filtering of stochastic signals*. Wydawnictwa Naukowo-Techniczne, 1998.
7. T. Kinjo and K. Funaki, "On HMM speech recognition based on complex speech analysis," in *Proceedings of 32nd Annual Conference of IEEE on Industrial Electronics, IECON 2006*, pp. 3477–3480, 2006.

---

# Adaptive Selection of Feature Set Dimensionality for Classification of DNA Microarray Samples

Henryk Maciejewski

Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland  
Henryk.Maciejewski@pwr.wroc.pl

**Summary.** This work tackles a problem of building predictive models from results of DNA microarray experiments. Data analysis challenges related to high dimensionality of data and small number of samples usually available from such experiments are discussed. A method is proposed to adaptively select the right number of genes to be used as features for a predictive model in order to avoid overfitting which seems to be the major risk in microarray studies. The approach proposed is illustrated by a numerical example based on a gene expression profiles from two types of acute leukemia (data originally published by Golub).

## 1 Introduction

Recent developments of DNA microarray or RT-PCR technologies allow to study relationships between expression of virtually all genes of tested cells and factors such as disease, response to treatment, response to changing environmental conditions etc. [1], [7]. Several specific data analysis problems have been formulated based on results of microarray studies, ranging from *class discovery*, through *class comparison* to *class prediction*. Class discovery involves clustering studies whose aim is to discover groups of genes with similar expression profiles across the samples tested (i.e. co-regulated genes) [3], [16], [10]; or, alternatively, it involves discovery of groups of samples similar in terms of their expression profile [1]. *Class comparison* studies assume that class membership of samples tested is known prior to analysis (e.g., samples related to some disease vs. controls). The purpose of these studies is to identify genes with different expression across classes compared. These results allow to identify genes whose expression is most strongly related to classes and which possibly account for the difference between the classes compared. The purpose of *class prediction* studies is to build a classifier capable of predicting class membership of future, yet unknown samples based on their gene expressions. The classifier is to be built based on results of a microarray experiments in which samples of known classes are examined. It is important to note that the goal of this task and its real criterion of success is high predictive accuracy of the classifier demonstrated for *new* samples.

This work concentrates on the problem of class prediction. This is a challenging task considering that microarray data usually contains relatively few samples of very high dimensionality. In the following section problems related to building a classifier from very high dimensionality data are discussed with particular emphasis given to the risk of overfitting and the problem of verification of predictive performance for future data. Next, a method is proposed to adaptively control dimensionality of the classifier feature set in order to hit a balance between classifier complexity and simplicity thus minimizing expected prediction error for future data. This is later illustrated by a numerical example based on acute leukemia microarray study, originally published in [7].

## 2 Problems of Classification in High Dimensionality Spaces

Trying to build a successful class prediction model based on data from microarray experiments is a difficult task due to very high dimensionality of data obtained and relatively small number of samples available. The challenges related to this are discussed in this part.

Let us introduce the following notation to represent results of a microarray study. Let  $x_i, y_i, 1 = 1, 2, \dots, n$  denote data related to the  $n$  samples tested in a microarray study, where  $x_i \in R^d$  represents gene expressions from sample  $i$  and  $y_i \in C = \{c_1, c_2, \dots, c_g\}$  denotes class membership associated with the sample  $i$ . Note that in the problem of class prediction outlined before the class membership  $y_i$  for each  $x_i$  is known prior to analysis. Often a binary problem is considered where  $y_i \in \{c_1, c_2\}$ . It should be noted that microarray studies typically produce data sets with  $d \sim 10^3$  to  $10^4$  (which is the number of genes observed in one DNA chip), while the number of samples tested  $n \sim 10^2$  or even fewer. In other words, microarray studies define an ill-formulated problem of classification, where  $d \gg n$ , while standard approaches to predictive modeling require that  $d \ll n$ .

Looking at this problem through the eyes of statistical decision theory [8], we can see data points  $\{x_i, y_i\}, i = 1, 2, \dots, n$  as  $n$  samples from random variables  $X \in R^d$  and  $Y \in C$ , with the joint probability distribution  $P(X, Y)$ . The problem of class prediction formally means that we are looking for a prediction model  $f : R^d \mapsto C$  minimizing the expected prediction error

$$EPE = E[L(Y, f(X))] \tag{1}$$

where  $L$  denotes the loss function penalizing wrong classifier decisions, which can be defined e.g.:

$$L(Y, f(X)) = \begin{cases} 1 & \text{for } Y \neq f(X) \\ 0 & \text{for } Y = f(X) \end{cases} \tag{2}$$

Obviously, in reality  $P(X, Y)$  is not known and hence the expected prediction error has to be approximated by empirical risk defined as  $\frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i))$ .

Considering the well know fact [8] pertaining to binary classification that in  $d$  dimension  $d + 1$  points can be always perfectly separated by a simple linear classifier, we can realize that for microarray data ( $\{x_i, y_i\}, i = 1, 2, \dots, n$  where  $d \gg n$ ), one can always obtain zero empirical risk for the training data. However such models will not guarantee good predictive performance for new data. This illustrates the major challenge related to classification given small number of points in high dimensional space, which is overfitting: meaning small prediction error for the training set of samples and high prediction error for test (i.e. new) data. These ideas justify the following key requirements that a successful procedure to build a predictive model from microarray data will have to meet:

- The method will have to significantly reduce dimensionality of points  $x_i$  in order to reduce the effect of overfitting and thus allowing for better generalization of the model for new data. However, dimensionality reduction which is a means to simplify the model can bring another risk related to the model being too simple to capture the relationship between  $X$  and  $Y$  [13]. Thus the right dimensionality of the vector of features obtained from vectors  $x_i$  will have to realize the balance between (over)complexity and (over)simplicity of the model. The objective function should measure predictive performance of the classifier for the test (new) data.
- The model building procedure will have to rigorously estimate expected performance of the predictive model for independent data. This problem is especially acute for microarray studies where the overall number of samples tested is relatively small. Thus data reuse methods such as cross-validation techniques have to be used in order to estimate expected prediction error for new samples.

The next section outlines an approach to class prediction using data driven, adaptive selection of the *right* dimensionality of the model, maximizing the predictive performance for new data, as measured by the means of data reuse techniques.

### 3 Adaptive Selection of Feature Set Dimensionality

In order to reduce dimensionality of points  $x_i$  prior to teaching a classification model, probably the most widely used method is *gene selection*. It typically involves one of two following approaches:

- An arbitrarily chosen number of genes which exhibit the most different expression across classes compared are selected as features. To do this, genes are first ranked using some statistic (such as t- or Wilcoxon-statistic, or F-statistic for the case of more than 2 groups) or heuristic (such as signal-to-noise ratio) measure of dissimilarity of expression across group. Then e.g. 50 top genes are used as features (see e.g., [7]).
- Alternatively, in order to avoid an arbitrary choice of the number of genes, one can select these genes as features that pass a statistical test of dissimilarity of expression across classes compared. More specifically, genes for which the p-value of a corresponding t- or Wilcoxon-test (or F-test for multi-class

problem) falls below, say, 5% threshold are used as features, the remaining genes are discarded. Often, in order to reduce the high risk of false positive outcomes related to the multiple tests effect, family-wise correction is applied (such as e.g. Bonferroni correction, [2]).

Both approaches (and their variants) imply that gene / feature selection is done independently of the classifier building stage of analysis.

We now formulate an approach that aims to use performance of the predictive model as a measure of informative value of the set of features used for building the model. This can be later used to select the optimal feature set. Details of this are given in the following procedure. We use notation introduced in Section 2, and concentrate on the binary classification problem ( $C = \{c_1, c_2\}$ ), which can be extended to multi-class problems by using F-statistic in Step 2.

- Step 1. Select a candidate dimensionality of feature set  $d^* : 1 \leq d^* \leq n - 2$ .
- Step 2. Select the top  $d^*$  genes using Wilcoxon test of similarity between groups. Specifically, for each gene  $k$ , showing expression  $x_i^k$  in sample  $x_i$  compute the value of the Wilcoxon rank test comparing the groups  $\{x_i^k : y_i = c_1\}$  versus  $\{x_i^k : y_i = c_2\}$ ; select  $d^*$  genes that realize the  $d^*$  highest values of the test statistic.
- Step 3. For  $i = 1, 2, \dots, n$ , compute the classification model  $f^{-i}$  by fitting it to data with the  $i$ th sample removed. Compute the following cross-validation estimate of the classifier prediction error EPE under  $d^*$  features as: [Hastie]

$$CV_{d^*} = \frac{1}{n} \sum_{i=1}^n L(y_i, f^{-i}(x_i)) \quad (3)$$

- Step 4. Repeat steps 1-3 for (all) different values of the candidates  $d^*$  in the range  $[1, \dots, n - 2]$ . Choose the value  $d^*$  that minimizes the  $CV_{d^*}$  (denote  $d_{min}$ ). In case the minimum value of  $CV_{d^*}$  is realized by more than 1 value of  $d^*$ , select the smallest winner (along with Occam razor machine learning paradigm).
- Step 5. Train the model  $f$  by fitting it to complete data of  $d_{min}$  dimensions. This model minimizes the estimated EPE.

It should be noted that the cross-validation scheme used in Step 3 realizes approximately unbiased estimate of EPE ([8]), while possible alternative approaches of k-fold cross-validation are biased, but possibly realize smaller variance. However, it seems that for microarray studies with very small number of samples (below 50?), bias in k-fold cross-validation due to even smaller number of training samples can lead to overestimates of prediction error [8]. This justifies the choice of the cross-validation scheme used in Step 3, despite its computational cost (i.e. model needs to be fitted to data  $n$  times for each fixed value of  $d^*$ ).

Although the method most widely used of ranking genes according to dissimilarity of their expression between classes seems to be the t-test (or similar means-comparison methods), in Step 2 application of rank-test has been proposed. Our studies have shown that rank-tests outperform means-comparison

tests in terms of informative value of genes selected (see e.g., [11]); some other arguments against application of t-tests have been reported in literature (see e.g. [12]).

## 4 Numerical Example

The method presented in the preceding section will be illustrated by a numerical study performed on the data originally published by Golub [7]. The task is to build a classifier to distinguish between two types of blood cancer: acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) based on patient's gene expression pattern. The data contains 72 samples, 25 samples of AML and remaining 46 of the ALL class.

Before the procedure described in the preceding section can be applied, data preprocessing has to be done, which involves:

- Normalization of samples from microarray in order to control / remove systematic variability related to different experimental conditions (such as unequal average brightness of chips). To do this, all samples  $x_i$  were multiplied by rescaling factors provided by Golub.
- Removing genes with mostly missing signal according to oligo microarray detection indicators (Absent, Present, Marginal) of quality of expression (signal) for each gene. In this study, we removed genes for which the number of Absent indications exceeded 30 out of 72 samples analyzed. After this procedure, 1707 genes remain for further analysis (out of the original 7071 genes; these numbers do not include Affymetrix house keeping probes, which were earlier removed from the study).

For the purpose of this example, the procedure shown in Section 3, has been repeated for the candidate set of feature set dimensions  $d^* \in \{5, 10, 20, 50, 70\}$ .

For example, for  $d^* = 20$  the following genes were selected in Step 2 (identified here by Affymetrix gene accession number): X62654\_rna1\_at, L09209\_s\_at, M16038\_at, M11722\_at, X61587\_at, M31523\_at, M92287\_at, M22960\_at, M62762\_at, X17042\_at, X07743\_at, J05243\_at, M31211\_s\_at, M11147\_at, X64072\_s\_at, M93056\_at, X59417\_at, HG3494-HT3688\_at, M63959\_at, HG1612-HT1612\_at.

The classification model fitted to data was multilayer perceptron, with  $d^*$  input neurons and 1 hidden layer with 1 neuron (the tool used to train the model was SAS Enterprise Miner neural network). For each value of  $d^*$  the total number of misclassification errors was observed for test data over  $n$  runs of the cross-validation procedure (this gives the sum in the expression (3) defining  $CV_{d^*}$ ). Also, training data errors were recorded. Results are shown in Table 1.

Note that for a given  $d^*$ ,  $\text{Test}_{ALL-AML} + \text{Test}_{AML-ALL}$  gives the sum term (numerator) in formula (3), hence we can directly estimate the  $EPE$  (1) by (3), e.g., for  $d^* = 20$  the  $CV_{d^*} = 0.028$ , and is the best model out of the tested models.

These results also clearly illustrate the problem of model overfitting (large values of  $d^*$ ), and the effect of a too simple model, not able to reflect the relationship between  $Y$  and  $X$  (small values of  $d^*$ ).

**Table 1.** Prediction errors vs. number of genes selected as features (dimensionality of model). Train column gives number of errors for training data, Test columns give number of errors observed for test data over  $n$  runs of cross-validation.

Features ( $d^*$ )	Train	Test <sub>ALL-AML</sub>	Test <sub>AML-ALL</sub>
5	0	2	4
10	0	1	2
20	0	1	1
50	0	2	1
70	0	6	2

## 5 Conclusions

It is important to note that the goal of class prediction task and its real criterion of success is high predictive accuracy of the classifier built demonstrated for new samples. Considering high dimensionality of microarray data together with small number of samples available, the task of building a predictive model which generalizes well to new data seems very challenging. In this work, we discuss a method to adaptively select right dimensionality of the feature set that offers a compromise between overfitting and over simplifying the model by gene selection. The method may help build models based on limited experiment data that have known predictive performance for new data. Although no numerical data reuse procedures can give more trustworthy estimates of predictive performance for new data than new data itself (possibly from another experimental site), results such as shown here can contribute to building more robust class prediction models. Being able to achieve this will help turn microarray technology into a practical tool for medical diagnosis, determination of risk groups or prediction of response to treatment (as pointed out in [12]).

## References

1. Bittner M, Meltzer P, Chen Y (2000) Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature* 406:536-540
2. Dudoit S, Shaffer J, Boldrick J (2002) Multiple Hypothesis Testing in Microarray Experiments. UC Berkeley Division of Biostatistics Working Paper Series, Paper 110.
3. Eisen M, et al. (1998) *Proc. Natl. Acad. Sci. USA* 95:14863-14868
4. Everitt B (1980) *Cluster Analysis, Second Edition*. Heineman Educational Books Ltd., London
5. Ewens W, Grant G (2001) *Statistical Methods in Bioinformatics*. Springer, Berlin Heidelberg New York
6. Faller D, et al. (2003) *Journal of Computational Biology* 10:751-762
7. Golub T, et al. (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531-537.
8. Hastie T, Tibshirani R, Friedman J (2002) *The Elements of Statistical Learning. Data Mining, Inference and Prediction*. Springer, Berlin Heidelberg New York

9. Hoffmann R, Seidl T, Dugas M (2002) Profound effect of normalization on detection of differently expressed genes in oligonucleotide microarray data analysis. *Genome Biology*
10. Maciejewski H, Jasinska A (2005) Clustering DNA microarray data. *Computer recognition systems CORES 05*, Springer *Advances in Soft Computing*
11. Maciejewski H, Konarski L (2007) Building a predictive model from data in high dimensions with application to analysis of microarray experiments. *DepCoS - REL-COMEX*, IEEE Computer Society Press
12. MAQC Consortium [Shi L. et al.] (2006) The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nature Biotechnology* 24
13. Markowitz F, Spang R (2005) Molecular diagnosis. Classification, Model Selection and performance evaluation, *Methods Inf. Med.* 44:438-443
14. Quackenbush J (2001) *Nature Reviews Genetics* 2:418-427
15. Shannon W, Culverhouse R, Duncann J (2003) *Pharmacogenomics* 4:41-51
16. Tamayo P, et al. (1999) *Proc. Natl. Acad. Sci. USA* 96:2907-2912



---

# Human Lips Recognition

Michał Choraś

Image Processing Group, Institute of Telecommunications  
University of Technology & Life Sciences  
S. Kaliskiego 7, 85-791 Bydgoszcz, Poland  
chorasm@utp.edu.pl

**Summary.** In many cases human identification biometrics systems are motivated by real-life criminal and forensic applications. Some methods, such as fingerprinting and face recognition, proved to be very efficient in computer vision based human recognition systems. In this paper we focus on developing computer recognition system that would be used to identify humans on the basis of their lips. We consider lips' shape and color features in order to determine human identity.

## 1 Introduction

Biometrics methods of human identification have gained much attention recently, mainly because they easily deal with most problems of traditional identification. In biometrics human identification systems users are identified by who they are, not by something they have to remember or carry with them.

There are many well known implemented methods of human identification (face, iris, retina, etc.), but still novel and innovative solutions are proposed. Some of the new biometrics modalities are: human scent recognition, EEG biometrics, skin spectroscopy, knuckles texture, finger-veins and finger-nails recognition [1][2][3][4].

Currently, new and unusual prints left in the crime-scene are taken into account by the police and forensic specialists. Such novel procedures include earprints, noseprints, forehead-prints as well as shoeprints [5][6][7].

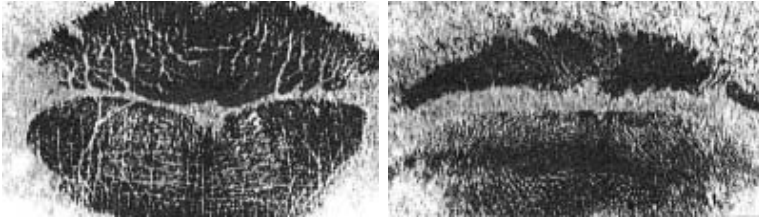
One of the most interesting emerging method of human identification, which originates from the criminal and forensic practice (*cheiloscopy*), is human lips recognition [5].

The fact that lip features may be unique for humans have been confirmed by Yasuo Tsuchihashi and Kazuo Suzuki in their studies at Tokio University (1968-1971). They examined 1364 subjects form 3 to 60 years of age of both genders. Their research proved that lip characteristics are unique and unchangeable (stable) for each examined person [5][8].

Lip prints characteristics have been successfully used in forensics experts and criminal police practice. Such approach to human identity confirmation is used by the police and have been included as a subdiscipline of dactyloscopy.

While examining human lips characteristics, most often, the anatomical patterns on the lips are taken into account. For example the pioneer of *cheiloscopy* and *otoscopy* in Poland, professor J. Kasprzak defined 23 lip patterns [5]. Such patterns (lines, bifurcations, bridges, pentagons, dots, lakes, crossings, triangles etc.) are similar to fingerprint, iris or palmprint patterns. The listed patterns of human lips have been available at

[http://www.kryminalistyka.fr.pl/crime\\_cheiloskopia.php](http://www.kryminalistyka.fr.pl/crime_cheiloskopia.php).



**Fig. 1.** Lip-prints samples used in forensic evidence expertise [5]

However, in image analysis computer recognition system, such features cannot be used since it is hard to extract them from acquired images. Therefore in our approach we do not use lipprints' features, but we focus on characteristics extracted from lip/face images. Such images can be acquired from a certain distance by a standard camera in a controlled or uncontrolled environment.

Automated lips recognition have not been extensively researched so far, but some very promising results were achieved by HMM and PCA methods [9].

In our research we decided to examine the possibility of recognizing lips from face images. We designed computer vision application to recognize humans on the basis of their extracted lips' shape and color features.

## 2 Lip Recognition

In our approach we calculate color features of the masked out lips and we merge them with shape features of the binarized lips. We calculate color statistics and moments as well as a set of standard geometrical parameters and the moments of Hu and Zernike.

### 2.1 Lip Detection

In the first step lips are detected from face images. Then we perform segmentation, binarization and size normalization. Most of known lips detection methods proposed so far, had been designed for speech recognition and tracking [10][11][12].

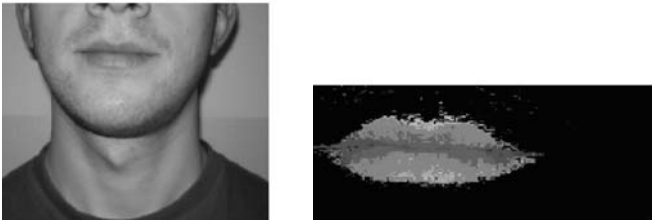
We modified a method based on color discriminates proposed by Kubanek [13][15]. In our case color discriminates are calculated not only in the *RGB* space, but in the *HSV* and *YUV* color spaces as well.



**Fig. 2.** Sample lower face images from our database (3 images of the same person)

The sample results of our lips detection algorithm from lower face images are presented in Figures 3 and 4.

So far our algorithms work well for face images captured especially for lip recognition project (lower face only). Unfortunately, we struggle with satisfactory lips detection on other datasets, especially face images from surveillance cameras.

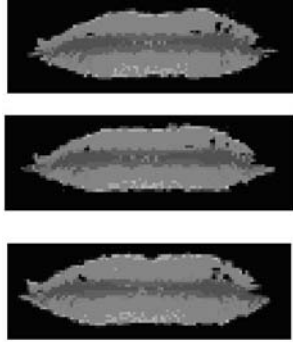


**Fig. 3.** Example of lower face and extracted corresponding lips area

### 2.2 Lips Shape Feature Extraction

After lips detection stage, shape features of the binarized lips images (Figure 4) are calculated.

We decided to calculate geometrical parameters of detected lips area. Shape feature vector used in our system consists of the following features:



**Fig. 4.** Examples of extracted lip shapes from face images

- central moments
- 7 Hu moments
- Zernike moments up to 3rd order
- Malinowska ratio
- Blair-Bliss
- Danielsson ratio
- Haralick ratio
- Lp1 ratio
- Lp2 ratio
- Feret ratio

### 2.3 Lip Color Recognition

We calculate statistical color features in three types of color spaces: *RGB*, *HSV* and *YUV*. Features are calculated separately for each channel in the used color spaces. Moreover, we use the MONO channel of the monochromatic image.

Color feature vector consists of the following features:

- maximum density - number of pixels of the most common intensity in the image
- minimum density - number of pixels of the least common intensity in the image
- mean given by  $m_i = \frac{1}{N} \sum_{j=1}^N p_{ij}$  where  $i$  and  $j$  are the pixel coordinates
- variance given by  $\sigma_i = \left( \frac{1}{N} \sum_{j=1}^N (p_{ij} - m_i)^2 \right)^{\frac{1}{2}}$  where  $i$  and  $j$  are the pixel coordinates
- slantity given by  $s_i = \left( \frac{1}{N} \sum_{j=1}^N (p_{ij} - m_i)^3 \right)^{\frac{1}{3}}$ .

Calculated lips color features in specific color spaces are presented in Figure 5. Some features may not be calculated within certain color spaces (marked with dash).

	COLOR CHANNEL									
	R	G	B	MONO	H	S	V	Y	U	V
Max density	1	5	9	12	16	20	24	-	-	-
Min. density	-	-	-	14	-	-	-	-	-	-
Mean	2	6	10	15	17	21	25	28	30	32
Variance	3	7	11	-	18	22	26	29	31	33
Slant	4	8	12	-	19	23	27	-	-	-

**Fig. 5.** The table presenting calculated lips color features in specific color spaces. Each calculated color feature is numbered. If the feature is not calculated in a given space it is marked with dash.

### 3 Experimental Results

In the experiments we used our own database of lower face images. We acquired 3 images per person. Database used in our experiments consists of 3 images from 38 people (114 images).

The feature vector describing lips consists of: 7 Hu moments, central moments, moments of Zernike, 10 standard geometrical parameters, statistical color features in *RGB*, *YUV* and *HSV* color spaces.

After feature extraction step we performed feature selection. After experiments we chose most discriminative and effective features. It turned out that after the selection step the recognition rate was improved of 10%.

Human lips recognition results before and after the feature selection step are shown in the Table 1.

**Table 1.** Lips recognition results

method	number of tests	correct acceptances	false rejections	Rank-1
all extracted features	38	25	13	66%
selected best features	38	29	9	76%

### 4 Conclusions

In this paper we presented pattern recognition methods and introductory results of automated human lips recognition system.

In our research integrated lip shape descriptors and color features had been used to determine human identity.

So far the major problem of our system is the lips detection step. We detect lips easily from face images captured especially for lip recognition project (lower face only). Unfortunately, we struggle with satisfactory lips detection on other datasets, especially face images from surveillance cameras.

However, we have achieved promising recognition results for well detected lips images, which motivates our further research in this area.

## References

1. Prabhakar S., Kittler J., Maltoni D., O’Gorman L., Tan T. (2007), Introduction to the Special Issue on Biometrics: Progress and Directions. *IEEE Trans. on PAMI*, vol. 29, no. 4, 513-516.
2. Goudelis G., Tefas A., Pitas I. (2005), On Emerging Biometric Technologies. In *Proc. of COST 275 Biometrics on the Internet*, 71-74, Hatfield UK.
3. Morales A., Ferrer M.A., Travieso C.M., Alonso J.B. (2007), A knuckles texture verification method in a transformed domain. In: *Proc. of 1st Spanish Workshop on Biometrics (on CD)*, Girona, Spain.
4. Choraś M. (2007), Emerging Methods of Biometrics Human Identification. In: *Proc. of ICICIC 2007 - Kummamoto, Japan*, IEEE CS Press.
5. Kasprzak J., Leczynska B. (2001), Cheiloscopy. Human Identification on the Basis of Lip Prints (in Polish). CLK KGP Press, Warsaw.
6. Kasprzak J., *Forensic Otopscopy (in Polish)*, University of Warmia and Mazury Press, 2003.
7. Huynh C., Chazal P.de, Flynn J., Reilly R.B.(2003), Automatic Classification of Shoeprints for use in Forensic Science. In: *Proc. of the Irish Machine Vision and Image Processing Conference*, Dublin, Ireland.
8. Tsuchihashi Y. (1974), Studies on Personal Identification by Means of Lip Prints. *Forensic Science*, 3:3.
9. Gomez E., Travieso C.M., Briceno J.C., Ferrer M.A. (2002), Biometric Identification System by Lip Shape. In: *Proc. of Carnahan Conference on Security Technology*, 39-42.
10. Ouyang H., Lee T. (2005), A New Lip Feature Representation Method for Video-based Bimodal Authentication. In: *Proc. of NICTA-HCSNet Multimodal User Interaction Workshop*, vol. 57, 33-37, Sydney, Australia.
11. Cetingul H.E., Yemez Y., Erzin E., Tekalp A.M. (2005), Robust Lip-Motion Features for Speaker Identification. In: *Proc. of IEEE Conf. on Acoustics, Speech, and Signal Processing - ICASSP 2005*, 509-512, Philadelphia, USA.
12. Cetingul H.E., Yemez Y., Erzin E., Tekalp A.M. (2006), Discriminative Analysis of Lip Motion Features for Speaker Identification and Speech-Reading. *IEEE Transactions on Image Processing*, 15:10, 2879 - 2891.
13. Kubanek M. (2006), Method of Speech recognition and Speaker Identification with Use Audio-Visual of Polish Speech and Hidden Markov Models. In: K.Saeed et.al. (Eds.) *Biometrics, Computer Security Systems and Artificial Intelligence Applications*, 45-55, Springer NY.
14. Kubanek M. (2005), A method of polish audio-video speech recognition based on Hidden Markov Models, PhD Thesis, Czestochowa University of Technology, 2005.
15. Kubanek M. (2005), Technique of Video Features Extraction for Audio-video Speech Recognition System. *Computing, Multimedia and Intelligent Techniques*, 1:181-190.

---

# Model Based Vehicle Extraction and Tracking for Road Traffic Control

Bartłomiej Płaczek and Marcin Staniek

Faculty of Transport, Silesian University of Technology Krasińskiego 13, 40-019  
Katowice, Poland

{bartlomiej.placzek,marcin.staniek}@polsl.pl

**Summary.** The paper presents a method for extracting and tracking vehicles in a video sequence. This scheme combines background updating and edge detection algorithms with model based object tracking. It enables a robust mapping of vehicles trajectories and reliable determination of occupancy of detection zones. The devised method is suitable for application in video sensors for road traffic control systems.

## 1 Introduction

Modern traffic control algorithms require reliable and real-time traffic information, e.g. [2]. Most of them utilize traffic characteristics extracted for defined regions - so called detection zones, where passing vehicles are counted, occupancy is detected or other measurements are performed (e.g. velocity).

In this field the video detection becomes more and more popular because it is cost effective and offers a number of advantages such as the ability to track vehicles and provide complex data on road traffic [13]. For conventional video detection systems, detection zones are defined as regions of image and only two-dimensional image features are considered in the detection algorithm. It means that extracted objects (vehicles) are treated as two-dimensional. This manner of processing leads to detection errors: recognized dimensions of vehicles (length and width) are bigger than their real dimensions; high vehicles can occupy detection zones localized behind them or in neighbouring traffic lanes [9]. These errors have a great impact on detection performance, especially when the camera is not installed at optimal location - appropriately high, directly above the lane.

The rest of this paper is organized as follows. Section 2 includes an overview of video-detection methods. The model-based method is presented in section 3. Section 4 introduces the vehicle extraction method. In Section 5 the scene and vehicle model is described. Section 6 describes proposed vehicle detection and tracking algorithm. The experiment results are shown in section 7 and finally, conclusions are drawn.

## 2 Related Works

Numerous vehicle detection algorithms using image processing techniques have been reported so far. One of the most widespread methods uses background

subtraction [15]. The algorithms based on background subtraction need additional operations that enable to cope with problems concerning automatic background extraction and updating. Main advantage of this method is the capability of moving vehicles detecting as well as stopped ones, which is crucial for analysis of intersection scenes.

Vehicles in motion can be detected by image differencing algorithms [3]. The major drawback of this simple method is that results strongly depend on the time interval between successive frames and vehicles velocity. Vehicles may remain undetected if they move very slowly in the camera field of view. Similar problems are encountered for optical flow methods [1]. Moreover, optical flow algorithms have a large computational cost and fail when pixel value variations are insignificant for image regions occupied by vehicles. However, this method allows for the determination of motion direction and therefore has been used for vehicles tracking in many applications.

Feature-based detection methods [8], [19], [17] include edge detection techniques, corner detection, texture analysis, harr-like features, colour features, etc. These methods are designed to find image features that are typical of vehicles. Feature extraction has been used for both stationary and moving vehicles detection as well as for objects tracking. In [12] neural networks have been proposed for detection of vehicles. A neural detector requires a representative set of images to perform a training procedure, which is time-consuming.

Model-based methods utilize shape models to recognize vehicles in the image [4]. Using 3-D models partially occluded vehicles can be correctly detected [7]. However, algorithms of this type have higher computational complexity.

Detection methods in many cases constitute an element of tracking procedures. Besides finding the current position of a vehicle, movement analysis is necessary to track the vehicle in a video sequence. For this purpose various methods have been used: optical flow estimation [6], template matching [5], rule-based approach [11], Kalman filtering [18], active contours [16], etc.

### 3 Proposed Method

In this study a model-based method for vehicle extraction and tracking is proposed. This method is intended for implementation in video sensors for road traffic control systems. The method uses 3-D models of vehicle and road scene with marked detection zones (virtual loops). The 3-D models map the shape of vehicle, its position in the scene as well as the position of camera and detection zones. The detection zone is defined in the road scene model as a set of points on the road plane. Occupancy factors for the detection zones are evaluated by analysing the routes of the moving models of vehicles.

Vehicles models are updated to reflect current parameters of the traffic stream. The model updating is performed using vehicle extraction results and vehicle movement characteristics. Background subtraction and edge detection techniques are used for vehicle extraction. Movement characteristics of vehicles are



determined by a template matching algorithm. The template is generated for a given vehicle shape and position, using algorithms of three dimensional computer graphics.

### 4 Vehicle Extraction

The proposed background recognition algorithm includes short-term and long-term updating procedures that enable to cope with illumination changes and shadows of stationary objects. Short-term updating is executed for each video frame  $F_m$ , according to formulas:

$$\begin{aligned}
 B_m &= \gamma(x, y)I_m(x, y) + [1 - \gamma(x, y)] B_{m-1}(x, y), \\
 \gamma(x, y) &= \begin{cases} 0, & |F_m(x, y) - B_{m-1}(x, y)| > t_1 \\ 0.1, & |F_m(x, y) - B_{m-1}(x, y)| \leq t_1 \end{cases} \quad (1)
 \end{aligned}$$

where  $m = 0, 1, \dots$  and  $B_m$  stands for background of frame  $F_m$ . Threshold value  $t_1 = 10$  was set experimentally.

Long-term updating is performed for periods of  $T$  frames. Thus, this operation takes into account frames denoted as  $F_m$ , where  $m = m' + n$  and  $n = 0, 1, \dots, T$ . Initialisation of this procedure is done for  $n = 0$ , initial values are:  $m' = m$  (index of the current frame),  $S_0(x, y, c) = 0$ ,  $V_0(x, y, c) = 0$ ,  $Z_0(x, y) = 0$ , where  $(x, y)$  are pixel coordinates (values are set for all pixels) and  $c \in \{0, 1, 2\}$ . During long-term updating period the following computations are being made:

$$\begin{aligned}
 S_n(x, y, c) &= \begin{cases} S_{n-1}(x, y, c) + F_{m'+n}, & F_{m'+n} \in [c\alpha; (c+1)\alpha] \\ S_{n-1}(x, y, c), & \text{else} \end{cases} \\
 V_n(x, y, c) &= \begin{cases} V_{n-1}(x, y, c) + 1, & F_{m'+n} \in [c\alpha; (c+1)\alpha] \\ V_{n-1}(x, y, c), & \text{else} \end{cases} \\
 Z_n(x, y) &= \begin{cases} Z_{n-1}(x, y) + 1, & |F_{m'+n} - B_{m'+n-1}(x, y)| > t_1 \\ Z_{n-1}(x, y), & |F_{m'+n} - B_{m'+n-1}(x, y)| \leq t_1 \end{cases} \quad (2)
 \end{aligned}$$

where  $n = 1, 2, \dots, T$ , and  $\alpha = 256/3$ . New value of background pixels are computed for the last frame in update period ( $n = T$ ):  $B_{m'+T}(x, y) = \beta(x, y)A(x, y) + [1 - \beta(x, y)]B_{m'+T}(x, y)$ ,  $A(x, y) = S_T(x, y, c')/V_T(x, y, c')$ , where value  $c'$  is chosen to fulfil the following condition:  $V_T(x, y, c') = \max_c \{V_T(x, y, c)\}$  and

$$\beta(x, y) = \begin{cases} 0, & Z_T(x, y) \leq t_2 \\ 0.5, & t_2 < Z_T(x, y) \leq t_3 \\ 1, & Z_T(x, y) > t_3 \end{cases} \quad \text{Threshold values } t_2 = 0.125T, t_3 = 0.25T$$

were empirically determined.

In the discussed algorithm a grey-scale morphological operator is used for edges detection. Pre-processing includes median filtering. After median filtering, erosion and dilatation are executed in parallel. In the next step subtraction is performed for images obtained as results of erosion and dilatation. Pixels values obtained after morphological transformations (edge image  $E(x, y)$ ) are binarized. This operation produces a binary map of edges  $E'$ . Edges are detected in frame  $F_m$  as well as in background  $B_m$ . Thus, two binary images are obtained:  $EF'_m$

for frame and  $EB'_m$  for background, where value 1 denotes edge presence for a given pixel. Finally, objects map  $O_m$  is computed as a result of vehicles extraction procedure. Values of elements  $O_m(x, y) \in \{0, 1, 2\}$  for all  $(x, y)$  are determined in three steps: 1)  $O_m(x, y) = 0$ ; 2)  $O_m(x, y) = 1$  if  $|F_m(x, y) - B_m(x, y)| > t_1$ ; 3)  $O_m(x, y) = 2$  if  $EF'_m(x, y) = 1$  and  $EB'_m(x, y) = 0$ . Thus, value 2 in the mask  $O_m$  indicates edge elements of vehicles (that are detected in the frame, but are not detected in the background); value 1 distinguishes elements of vehicles that do not belong to the background, value 0 denotes regions, where background is visible (no foreground object is present).

### 5 Road Scene and Vehicle Models

For the discussed model three different coordinate systems have been defined, as shown in fig. 1. The first  $(x, y, z)$  is a camera centred coordinate system, where  $z$  axis is the optical axis of the camera and  $x - y$  plane is parallel to the image plane. The second coordinate system  $(x', y', z')$  is road-fixed system, where the road plane is defined by  $x'$  and  $z'$ . Intersection point of  $z$  axis and  $x' - z'$  plane determines the origin of the road-fixed system. The third coordinate system  $(x'', y'', z'')$  is used for vehicle shape description, where  $y''$  is a rotation axis of the vehicle model and it is parallel to  $y'$  ( $x''$  and  $z''$  lie on  $x' - z'$  plane). Detection zone is modelled as a set of evenly spaced points on the road plane. For these detection points vehicle presence is checked to verify if a zone is occupied or not. In the presented approach detection area is a parallelogram. Thus, three vertices define a detection zone  $(x'_i, z'_i)$ ,  $i \in \{0, 1, 2\}$ . Pairs of vertices (0,1) and (0,2) determine two adjacent sides of the parallelogram. Coordinates of detection points  $P_{ij}$  are computed according to formula:

$$P_{ij} = (x'_0, z'_0) + i/Nx \cdot (x'_1 - x'_0, z'_1 - z'_0) + j/Nz \cdot (x'_2 - x'_0, z'_2 - z'_0). \quad (3)$$

Points  $P_{ij}$  are arranged in  $Nx + 1$  columns and  $Nz + 1$  rows:  $i = 0, 1, \dots, Nx$ ,  $j = 0, 1, \dots, Nz$ . Values of  $Nx$  and  $Ny$  depend on size of detection zone and its location in the image.

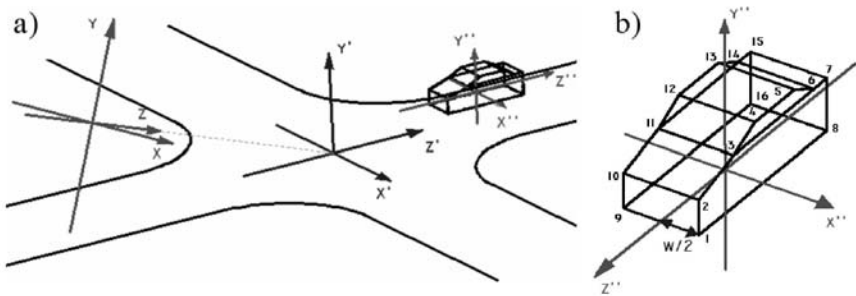


Fig. 1. 3-D models of scene and vehicle

Definition of the vehicle model comprises its width  $w$  (dimension along  $x''$  axis) and coordinates of eight vertices that describe the shape of the model on plane  $y'' - z''$ . This two-dimensional model can be written in form of a  $8 \times 2$  matrix:  $M = [m_{ij}]$ , where  $(m_{i0}, m_{i1})$  are coordinates of  $i$ -th vertex,  $i = 18$ .

The template for vehicle model is generated for a given vehicle position in the scene and rotation angle. Position is defined by vector  $L = [l_1, 0, l_2]^T$  that also determines the origin of the vehicle-fixed system in the road plane. 3-D vehicle model (fig. 1 a) is created in road-fixed coordinate system on the basis of matrix  $M$ . New coordinates of vertices are represented by rows of  $M'$  matrix. The vehicle model is rotated around  $y'$  axis and translated by the position vector. The next operation in the template generation algorithm is the calculation of viewing transformation [10], which consist of: conversion of road-fixed system to camera coordinate system, visibility calculations and transformation for perspective projection. As a result of this operation coordinates of pixels are obtained that correspond to visible vertices of the vehicle model.

The vehicle model (fig. 1 b) comprises pairs of vertices that are linked by edges. Edges in the template are generated linking pixels representing vertices of the model, which were determined in the previous operation. Bersenham's line drawing algorithm has been applied for selection of edge pixels. After edges drawing, dilation is performed to widen the edges and Smith's flood fill algorithm is used to mark pixels in regions bounded by edges. These bounded regions of template correspond to facets of the 3-D vehicle model. A two-dimensional template  $TE_m$  generated for the vehicle describes its shape and position in  $m$ -th frame using three pixel values  $TE_m(x, y)$ : value 0 denotes elements of empty region, 1 indicates elements that belong to vehicle regions and 2 stands for edge elements;  $x$  and  $y$  are coordinates defined for video frame. Template  $TE_m$  is matched with objects map  $O_m$ . On this basis the vehicle position, assumed in 3-D model, is compared with the position registered in the traffic scene.

## 6 Vehicle Detection and Tracking

Assigned vehicle shape and position in the video frame is verified by a template matching procedure. The procedure computes the correlation factors of template  $TE_m$  and objects map  $O_m$ :

$$D(TE_m, O_m) = \max_{(\delta x, \delta y)} \left\{ \frac{1}{3} \cdot \frac{\sum_x \sum_y R(TE_m(x, y), O_m(x + \delta x, y + \delta y), 1)}{\sum_x \sum_y L(TE_m(x, y), 1)} + \frac{2}{3} \cdot \frac{\sum_x \sum_y R(TE_m(x, y), O_m(x + \delta x, y + \delta y), 4)}{\sum_x \sum_y L(TE_m(x, y), 2)} \right\}, \quad (4)$$

$$R(te, o, a) = \begin{cases} 1, & te \cdot o \geq a \\ 0, & else \end{cases}, \quad L(te, a) = \begin{cases} 1, & te \geq a \\ 0, & else \end{cases}, \quad (5)$$

Translation vector  $(\delta x, \delta y)$  takes into account small changes of coordinates (few pixels). As a result the template is shifted into a new position, where correlation value is at its maximum. Vehicle model position in the scene model is updated for each video frame. A new vehicle model is created if a vehicle is recognized

in the initialisation zone and then the vehicle is tracked as long as it remains within the camera field of view.

Model initialisation procedure uses a pre-defined 3-D vehicle model located in the so-called initialisation zone. A template created for this model enables the tracking procedure to check if a new vehicle enters the initialisation zone. The tracking procedure is started if the correlation measure for this template exceeds threshold  $t_{ini}$ . Next tracking thread cannot be started until the template correlation value falls below the threshold  $t'_{ini}$  ( $t'_{ini} < t_{ini}$ ). This thresholding with hysteresis protects the system against multiple tracking of the same vehicle. A copy of the initialisation model is assigned to each vehicle entering the initialisation zone and it is then relocated during vehicle tracking.

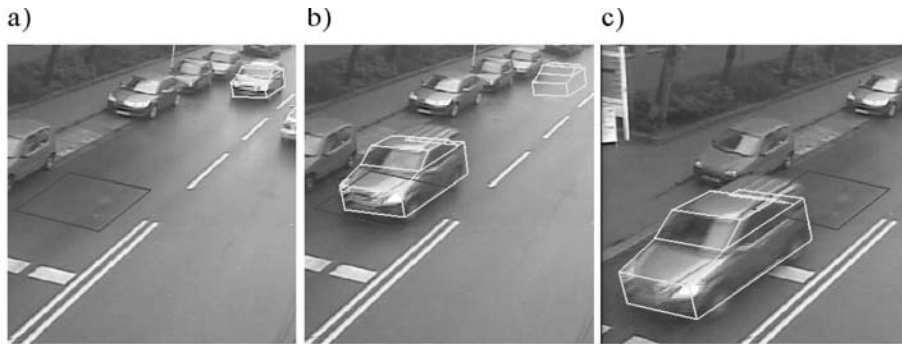
The tracking procedure consists of template matching, vehicle displacement determination for two successive frames, vehicle model relocation in the scene model and creation of a new template for the current position of the tracked vehicle. Displacement of the tracked vehicle is determined using cross correlation method. Cross correlation measure is computed using image region determined by template positioned in frame  $m - 1$ . This region of the previous frame is matched with frame  $m$  and best position is selected with respect to the cross correlation values. On this basis the relocation vector is determined for vehicle in 3-D road scene model and the vehicle model is shifted to a new position.

The vehicle model contains four vertices located on the road plane, which determine road area occupied by the vehicle. Detection zone is modelled as a set of points on the road plane.

Occupancy factor of the detection zone depends on the number of detection points, located in occupied area of the road. The occupancy factor calculation procedure uses an algorithm that enables to check if a given point of the detection zone lies in the area of the quadrilateral defined by the vehicle vertices [14]. Value of occupancy factor  $PNO/PN$  is computed, where  $PNO$  stands for the number of detection points in regions occupied by vehicles and  $PN$  denotes the number of all detection points that belong to the detection zone. If the occupancy factor  $PNO/PN$  is higher than an adopted threshold, the zone is occupied. The value of threshold is one of the configuration parameters of the detection zone.

## 7 Experimental Results

Experiments were carried out with video data of a crossroad approach. The tests were performed using a number of image sequences of traffic scenes taken in different lighting conditions and using various cameras. Figure 3 a shows a frame where the tracking process starts - a vehicle is detected within initialisation zone and a new vehicle model is created. Figure 2 b depicts a situation where vehicle is present in the detection zone. Occupancy of the zone is recognized on the basis of vehicle model position in the 3-D scene model. In figure 2 c the vehicle leaves the detection zone and the zone is correctly recognized as empty, although in the image the vehicle stays within the zone region.



**Fig. 2.** Vehicle tracking

The proposed model-based vehicle detection and tracking method eliminates a considerable number of false detection errors. False detections reduction is observed also when vehicles moving in neighbouring traffic lane (in the opposite direction) enter the area of detection zone in the image.

## 8 Conclusion

The proposed method uses 3-D models for extracting and tracking vehicles in a video sequence. In comparison with conventional video-detection systems, the number of detection errors is reduced due to integration of detection zones with the 3-D road scene model. Road regions occupied by vehicles are precisely determined by simple analysis of the 3-D traffic scene model. The results of tests using real-world videos show that the method gives reliable detection zone occupancy factors. The low computational complexity makes it suitable for implementation in road traffic control systems.

This research is a part of a project financed by the European Union and the Polish Ministry of Science and Higher Education.

## References

1. Adamski A, Bublinski Z, Mikrut Z, Pawlik P (2004) The image-based automatic monitoring for safety traffic lanes intersections. In: J. Piecha (ed) *Trans. on Transport Systems Telematics*. Wyd. Pol. Ćłl., Gliwice
2. Chen D, Li Z, Zhang L (2004) TCP, a traffic signal control algorithm based on knowledge and its simulation using RTE, *ITS Proc.*: 1033 - 1037
3. Fathy M, Siyal MY (1995) Real-time image processing approach to measure traffic queue parameters. *Vision, Image and Signal Processing Proc.* Vol. 142 No. 5: 297 - 303
4. Ferryman JM, Worrall AD, Sullivan GD, Baker KD (1995) A Generic Deformable Model for Vehicle Recognition. Vol.1: 127-136

5. Guan A, Peng Z (2006) An Improved Approach for Vehicles Detection and Tracking. 6th Int. Conf. on ITS Telecommunications Proc.: 306 - 309
6. Haag M, Nagel HH (1999) Combination of Edge Element and Optical Flow Estimates for 3D-Model-Based Vehicle Tracking in Traffic Image Sequences. International Journal of Computer Vision Vol. 35 No. 3: 295 - 319
7. Haag M, Nagel HH (1998) Tracking of complex driving manoeuvres in traffic image sequences Image and Vision Computing Vol. 16: 517-527
8. Hongliang B, Jianping W, Changpin L (2006) Motion and haar-like features based vehicle detection. 12th Int. Multi-Media Modelling Conf: 356-359
9. Inigo RM (1989) Application of machine vision to traffic monitoring and control. IEEE Trans. on Vehicular Technology Vol. 38 No. 3: 112 - 122
10. László SK (1995) Theory of Three Dimensional Computer Graphics. Publishing House of the Hungarian Academy of Sciences, Budapest
11. Lin SP, Chen YH, Wu BF (2006) A Real-Time Multiple-Vehicle Detection and Tracking System with Prior Occlusion Detection and Resolution, and Prior Queue Detection and Resolution. 18th Int. Conf. on Pattern Recognition Vol. 1: 828 - 831
12. Mantri S, Bullock B, Garrett J Jr (1997) Vehicle detection using a hardware-implemented neural net. Expert IEEE Vol. 12: 15 - 21
13. Michalopoulos PG (1991) Vehicle detection video through image processing: the Autoscope system. IEEE Trans. on Vehicular Technology Vol. 40 No.1: 21 - 29
14. Preparata FP, Shamos MI (1985) Computational geometry. Springer, Berlin
15. Siyal NY, Fathi M, Atiquzzaman M (2000) A Parallel Pipeline Based Multiprocessor System For Real-Time Measurement of Road Traffic Parameters. Real-Time Imaging Vol. 6: 241-249
16. Tai JC, Tseng SH, Po LC, Song KT (2004) Real time image tracking for automatic traffic monitoring and enforcement applications. Image and Vision Computing Vol. 22: 485-501
17. Tsai LW, Hsieh JW, Fan KC (2007) Vehicle Detection Using Normalized Color and Edge Map. Trans. on Image Processing Vol. 16 No.3: 850 - 864
18. Xie L, Zhu G, Wang Y, Xu H, Zhang Z (2005) Real-time vehicles tracking based on Kalman filter in a video-based ITS. Proc. Int. Conf. on Communications, Circuits and Systems Vol. 2: 883-886
19. Zhuang X (2004) Vehicle Detection and Segmentation in Dynamic Traffic Image Sequences with Scale-Rate, The 7th Int. IEEE Conf. on Intelligent Transportation Systems Proc: 570 - 574

---

# Optimization of Linear Fuzzy Gene Network Model Searching

A. Gintrowski<sup>1</sup>, E. Tkacz<sup>1</sup>, P. Kostka<sup>1</sup>, A. Wieclawek<sup>2</sup>, and U. Mazurek<sup>2</sup>

<sup>1</sup> Institute of Electronics, Division of Microelectronics and Biotechnology, Silesian University of Technology, Gliwice, Poland

{agintrowski, etkacz, pkostka}@polsl.pl

<sup>2</sup> Medical University of Silesia, Faculty of Pharmacy, Department of Molecular Biology, Sosnowiec, Poland

{awieclawek, umazurek}@slam.katowice.pl

**Summary.** Recently, almost uncontrolled technological progress allows so called high-throughput data collection for sophisticated and complex experimental biological systems analysis. Especially it concerns the whole cellular genome. Therefore it becomes more and more vital to suggest and elaborate gene network models, which can be used for more complete interpretation of large and complex data sets. The presented paper concerns modeling of yeast cell cycle. With the reference to the previously published papers concerning the same subject, our paper presents a significant improvement in calculation procedure leading to construction of an efficient algorithm for modeling of the mentioned cell cycle. Presented model concerns 12 selected genes and incorporates all possible types of interactions.

## 1 Introduction

Presented paper refers to the modeling of genes involved in the yeast cell cycle and can be considered as the important modification of the calculation procedure. The method of construction of fuzzy network model for the mentioned above medium has been described in details in [1] [2] [3]. However the exercise repeated based upon the recalled paper requires an extremely burdensome calculation procedure for which the detailed analysis will be presented in further sections, when the example of selected twelve genes analysis will be described.

### 1.1 Applied Notations

An efficient approach to construction of improved calculation procedure usually starts with suggestion of proper notations, which allows for brief formulation of important dependance. So we operate the notation referring to 4 inputs and just one output and taking into account only three values necessary for fuzzy rules construction for each input, i.e. low, medium and high:

$$r = [r_1, r_2, r_3] \tag{1}$$

- $r_1$  - output expression if input is Low
- $r_2$  - output expression if input is Medium
- $r_3$  - output expression if input is High

The example below will hopefully leave all the doubts concerning that important stage of suggested modifications:

- IF *Input* is Low THEN *Output* is High
- IF *Input* is Medium THEN *Output* is Medium
- IF *Input* is High THEN *Output* is Low

Such a linguistic expression, using suggested notation can be rewritten as follows:

$$r = [3, 2, 1].$$

### 1.2 Exhaustive Search of Linear Fuzzy Gene Network Models

We can create a gene expression vector in the way as:

$$G_i = [e_1, e_2, \dots, e_N] \tag{2}$$

where vector elements consist the values of measured expressions. Following that a proper matrix of expression data from analyzed microarray can be created as below:

$$M = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,N} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,N} \\ \vdots & & \ddots & \\ e_{M,1} & e_{M,2} & \cdots & e_{M,N} \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_M \end{bmatrix} \tag{3}$$

for which the next step requires proper data normalization:

$$\widehat{M} = \frac{\arctan(M)}{\frac{\pi}{2}} \tag{4}$$

At this stage it is necessary to perform data fuzzification and the creation of proper fuzzy sets is a crucial point for further analysis. It can be expressed using the following maneuver:

$$F_{low_{i,j}} = \begin{cases} -\widehat{e}_{i,j} & \widehat{e}_{i,j} < 0 \\ 0 & \widehat{e}_{i,j} > 0 \end{cases} \tag{5}$$

$$F_{medium_{i,j}} = 1 - |\widehat{e}_{i,j}| \tag{6}$$

$$F_{high_{i,j}} = \begin{cases} 0 & \widehat{e}_{i,j} < 0 \\ \widehat{e}_{i,j} & \widehat{e}_{i,j} > 0 \end{cases} \tag{7}$$

As the result of this process the three dimensional matrix of fuzzy sets is created:



$$F = [F_{low}, F_{medium}, F_{high}] \tag{8}$$

The analyzed output gene can be expressed, using all the previously introduced notations, as follows:

$$G_{out} = [e_{out_1}, e_{out_2}, \dots, e_{out_n}] = M \Big|_{\{out\} \times \{1, \dots, N\}} \tag{9}$$

then the fuzzyfied output gene should be consequently expressed as:

$$F_{out_{low}} = F_{low} \Big|_{\{out\} \times \{1, \dots, N\}} \tag{10}$$

$$F_{out_{medium}} = F_{medium} \Big|_{\{out\} \times \{1, \dots, N\}} \tag{11}$$

$$F_{out_{high}} = F_{high} \Big|_{\{out\} \times \{1, \dots, N\}} \tag{12}$$

$$F_{out} = [F_{out_{low}}, F_{out_{medium}}, F_{out_{high}}] \tag{13}$$

Discussed method requires analysis of all possible C combinations of k-input genes. Each of them can be described as follows:

$$F_{in_{low}} = F_{low} \Big|_{\{C\} \times \{1, \dots, N\}} \tag{14}$$

$$F_{in_{medium}} = F_{medium} \Big|_{\{C\} \times \{1, \dots, N\}} \tag{15}$$

$$F_{in_{high}} = F_{high} \Big|_{\{C\} \times \{1, \dots, N\}} \tag{16}$$

$$F_{in} = [F_{in_{low}}, F_{in_{medium}}, F_{in_{high}}] \tag{17}$$

Inference engine uses matrix *R* of *r* rule vectors for each input gene, discussed at the beginning of paper Eq.(1). To calculate the performance index *E*, as in the reference paper [1], proper summation procedures is required, and it can be expressed using the following set of formulas:

$$\widetilde{F}_{low} = \sum_{i=1}^k F_{in} \Big|_{\{C_i\} \times \{1, \dots, N\} \times \{R' |_{\{1\} \times \{i\}}\}} \tag{18}$$

$$\widetilde{F}_{medium} = \sum_{i=1}^k F_{in} \Big|_{\{C_i\} \times \{1, \dots, N\} \times \{R' |_{\{2\} \times \{i\}}\}} \tag{19}$$

$$\widetilde{F}_{high} = \sum_{i=1}^k F_{in} \Big|_{\{C_i\} \times \{1, \dots, N\} \times \{R' |_{\{3\} \times \{i\}}\}} \tag{20}$$

$$\tilde{F} = [\widetilde{F}_{low}, \widetilde{F}_{medium}, \widetilde{F}_{high}] \tag{21}$$

The two more stages to finish up the calculation procedure concern the defuzzification as well as denormalisation. Next set of equations describe the suitable calculations:

$$\hat{D} = \frac{\widetilde{F}_{high} - \widetilde{F}_{low}}{\widetilde{F}_{low} + \widetilde{F}_{medium} + \widetilde{F}_{high}} \tag{22}$$

$$D = \tan(\hat{D} \cdot \frac{\pi}{2}) \tag{23}$$

Finally we reach the stage when the calculating of models fitting coefficient is possible:

$$E = \frac{\sum_{i=1}^N (G_{out_i} - D_i)^2}{\sum_{i=1}^N (G_{out_i} - \overline{G_{out}})^2} \tag{24}$$

With this method, an exemplary analysis of yeast microarray with twelve genes, twenty examples each, executed with four inputs, requires  $12 \cdot (C_{11}^4 \cdot 27^4 + C_{11}^3 \cdot 27^3 + C_{11}^2 \cdot 27^2 + C_{11}^1 \cdot 27^1) = 2105006904$  inference operations. Considering at this stage even smallest time unit for each operation we obtain almost impractical method for linear fuzzy gene network model. It is necessary however, to underline at this moment that we accepted model, which should not be considered as too complicated as it has just 4 inputs for every twelve genes. Eventual calculations for one input more causes that calculations become totally impractical. In this section we will describe the important modifications, significantly simplifying

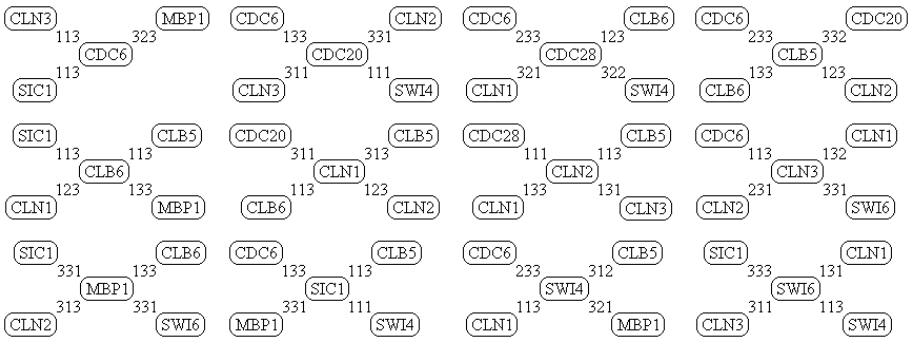


Fig. 1. Yeast gene network model

the whole calculation procedure. Let us start with comments concerning yeast gene network model presented on fig.1. Fig. 1 presents the mutual interactions between analyzed genes.

## 2 Modified Method for Gene Network Model Search

The idea of the optimization procedure is to limit the number of  $r$  vectors applied for inferences engine described above. We have applied the set of new  $r$  vectors for best rules of each fuzzy set calculated using modified performance index, which now is dependant on fuzzyfication. This is described as follows:

**Table 1.** Results of exhaustive search

Output	1st result		2nd result		3rd result		Inputs	Rules	
	Fit	Inputs	Rules	Fit	Inputs	Rules			Fit
SIC1	0,1983	CLB5	1 1 3	0,1998	CLB5	1 1 3	0,2051	CLB5	1 1 3
		SWI4	1 1 1		SWI6	3 1 3		CDC28	1 3 1
		MBP1	3 3 1		MBP1	3 3 1		MBP1	3 3 1
		CDC6	1 3 3		CDC6	1 3 3		CDC6	1 3 3
CLN1	0,0445	CLB5	3 1 3	0,0452	CLB5	1 3 3	0,0457	CLB5	1 1 3
		CDC20	3 1 1		CDC20	3 3 1		CDC20	3 1 1
		CLN2	1 2 3		CLN2	1 3 3		CLN3	1 2 1
		CLB6	1 1 3		CDC6	1 2 1		CLN2	1 3 3
CLN2	0,1189	CLB5	1 1 3	0,1214	CLB5	1 1 3	0,1281	CLB5	1 1 3
		CLN3	1 3 1		CLN1	1 2 3		CLN3	1 1 1
		CLN1	1 3 3		CLN1	1 3 3		CLN1	1 3 3
		CDC28	1 1 1		CDC28	1 1 1		SWI4	3 1 3
CLN3	0,264	SWI6	3 3 1	0,2784	SWI6	3 1 1	0,281	SIC1	1 1 1
		CLN1	1 3 2		CLN1	1 3 2		CDC20	3 3 1
		CLN2	2 3 1		CLN2	2 3 1		CW16	3 2 1
		CDC6	1 1 3		CDC6	1 1 3		CDC6	1 1 3
SWI4	0,2591	CLB5	3 1 2	0,2645	SIC1	2 1 2	0,2675	SIC1	2 1 2
		CLN1	1 1 3		CLB5	1 1 1		CLB5	1 1 1
		MBP1	3 2 1		CLN1	1 3 1		CLB6	1 1 1
		CDC6	2 3 3		MBP1	3 1 1		MBP1	3 1 1
SWI6	0,2906	SIC1	3 3 3	0,3105	SIC1	3 3 3	0,3107	CLN3	3 1 1
		CLN3	3 1 1		CLN3	3 1 1		CLN1	1 3 1
		CLN1	1 3 1		CLN1	3 1 3		CLB6	2 3 3
		SWI4	1 1 3		SWI4	1 1 3		MBP1	3 1 2
CLB5	0,1051	CDC20	3 3 2	0,1058	CDC20	3 2 2	0,1072	CDC20	3 1 2
		CLN2	1 2 3		CLN2	1 1 3		CLN2	1 2 3
		CLB6	1 3 3		CLB6	1 3 3		CLB6	1 1 3
		CDC6	2 3 3		CDC6	2 3 3		CDC6	2 3 3
CLB6	0,2059	SIC1	1 1 3	0,22	SIC1	1 1 3	0,2213	SIC1	1 1 3
		CLB5	1 1 3		CLN1	1 1 3		CLB5	1 2 3
		CLN1	1 2 3		CDC28	1 3 3		CLN1	1 1 3
		MBP1	1 3 3		MBP1	1 3 3		MBP1	1 3 3
CDC6	0,192	SIC1	1 1 3	0,1927	SIC1	1 2 3	0,1969	SIC1	1 1 3
		CLN3	1 1 3		CDC20	1 3 3		CLN3	1 1 3
		MBP1	3 2 3		CLN3	1 1 3		SWI4	1 1 1
					CDC28	1 1 1		MBP1	3 2 3
CDC20	0,1845	CLN3	3 1 1	0,2003	CLN3	3 1 1	0,2053	CLN3	1 3 1
		CLN2	3 3 1		CLN2	3 3 1		CLN1	3 1 1
		SWI4	1 1 1		SWI4	1 3 1		SWI4	1 1 3
		CDC6	1 3 3		CDC6	1 3 3		MBP1	3 3 3
CDC28	0,5798	CLN1	3 2 1	0,583	CLN1	3 3 1	0,5858	CLN1	3 1 1
		CLB6	1 2 3		CLB6	1 2 3		CLB6	1 2 3
		SWI4	3 2 2		SWI4	3 2 2		SWI4	3 2 2
		CDC6	2 3 3		CDC6	2 3 3		CDC6	2 3 3
MBP1	0,3185	SIC1	3 3 1	0,3257	SIC1	3 3 1	0,3268	SIC1	3 1 1
		SWI6	3 3 1		CLN2	3 1 3		SWI6	3 2 1
		CLN2	3 1 3		CLB6	1 3 3		CLB6	1 3 3
		CLB6	1 3 3		SWI4	3 3 1		SWI4	3 3 1

$$E_{low} = \sum_{i=1}^N |F_{out_{low}} - \widetilde{F}_{low}| \tag{25}$$

$$E_{medium} = \sum_{i=1}^N |F_{out_{medium}} - \widetilde{F}_{medium}| \tag{26}$$

$$E_{high} = \sum_{i=1}^N |F_{out_{high}} - \widetilde{F}_{high}| \tag{27}$$

Instead of calculation of great numbers of all the possible inferences we are able to limit calculations up to just a few best fitting performance indexes.

### 3 Results

The Table 2 presents the best performance indexes for consequently 5 to 25 best rules for each fuzzy set. It is possible to notice that with the increasing number of fuzzy rules the probability of agreement to exhaustive search method requires less calculation and the price for that is non-significant calculation time increase. The last column of the Table 2 is especially suggestive as it can be interpreted as follows: with just less than 5% of exhaustive method calculation time we are able to reach comparable results, considering the performance index.

**Table 2.** Comparing results

Gene	5 best rules time: 0,1%	10 best rules time: 0,39%	15 best rules time: 1,12%	20 best rules time: 2,55%	25 best rules time: 4,9%
SIC1	1	1	1	1	1
CLN1	24	11	1	1	1
CLN2	2	2	1	1	1
CLN3	236	18	18	18	5
SWI4	3359	1047	12	12	10
SWI6	293	23	1	1	1
CLB5	121	121	121	121	121
CLB6	7	7	2	1	1
CDC6	1	1	1	1	1
CDC20	4	4	1	1	1
CDC28	58579	12313	722	246	49
MBP1	14	1	1	1	1

Figures 2 and 3 presents comparison of obtained results for the same gene CLB5. In case of modified method it concerns the worst result obtained. To reach such a comparable result for the mentioned gene CLB5, only 4,9% of calculation time has been necessary, regarding exhaustive search.

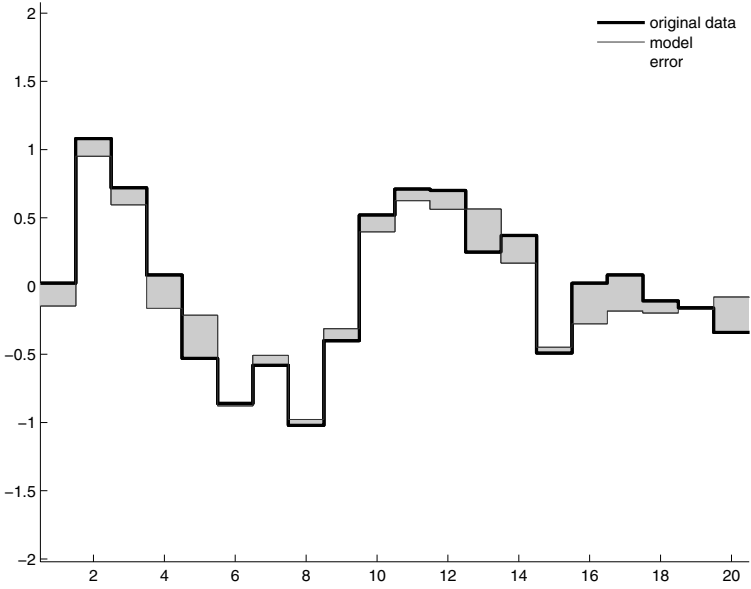


Fig. 2. Exhaustive search for CLB5; fit = 0,105

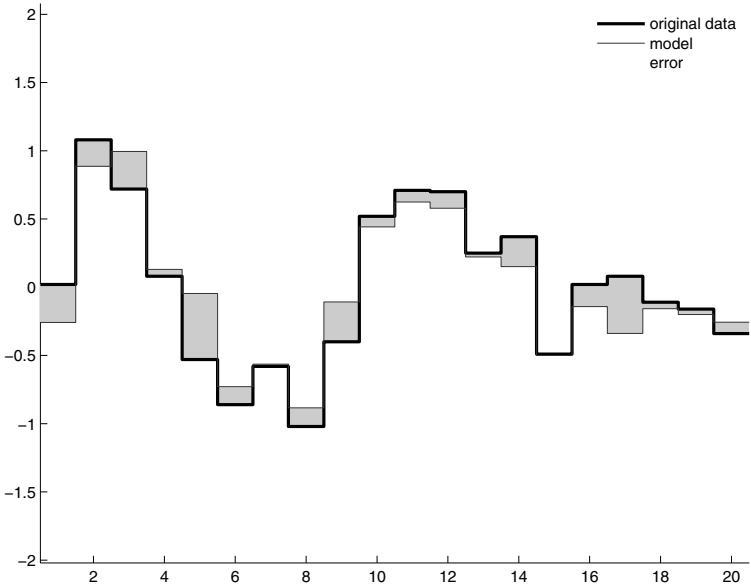


Fig. 3. Worst result of optimized method - search for CLB5; fit = 0,137

## References

1. Sokhansanj B, Fitch J, Quong J, Quong A, Linear fuzzy gene network models obtained from microarray data by exhaustive search, BMC Bioinformatics 2004, 5:108 doi:10.1186/1471-2105-5-108
2. Fitch J, Sokhansanj B "Genomic engineering: moving beyond DNA sequence to function", Proc IEEE 2000, 88:1949-1971(2000)
3. Griffin T, Gygi S, Ideker T, Rist B, Eng J, Hood L, Aebersold R, Complementary profiling of gene expression at the transcriptome and proteome levels in *Saccharomyces cerevisiae*, Mol Cell Proteomics 2002, 1:323-333

---

## Author Index

- Al Haj, Murad 19, 164, 179  
Al Modarresi, Mohammad T. 755  
Amato, Ariel 19, 164, 179  
Andreasik, Jan 718  
Augustyniak, Piotr 574
- Badr, Ghada 437  
Badura, Paweł 679  
Bator, Marcin 596  
Benmachiche, Karim 663  
Bican, Jakub 44  
Bielecka, Marzena 313  
Bieniecki, Wojciech 470  
Biernacki, Paweł 334  
Biesiada, Jacek 242  
Billard, Aude G. 663  
Bobrowski, Leon 348  
Borkowski, Dariusz 92  
Burda, Andrzej 802, 808  
Burduk, Robert 364
- Callejas, Zoraida 747  
Cardoso, Jaime S. 518  
Cardoso, Maria J. 518  
Carrasco-Ochoa, J. Ariel 27  
Chakraborty, Bhaskar 739  
Chaloupka, Josef 413  
Chmielewski, Leszek J. 596  
Choraś, Michał 838  
Choraś, Ryszard S. 340
- Drahanský, Martin 526  
Duch, Włodzisław 242  
Dulewicz, A. 613
- Einsele, Farshideh 429
- Flusser, Jan 44  
Forczmański, Paweł 3  
Frejlichowski, Dariusz 3
- Gacek, Adam 532  
Gargiulo, Francesco 705  
Gawlikowski, M. 558  
Gintrowski, A. 852  
Glogaza, Michał 763  
Głomb, Przemysław 117  
Golonek, T. 794  
Gomolka, Zbigniew 637  
González, Jordi 19, 164, 179, 739  
Goszczyńska, Hanna 621  
Grąbczewski, Krzysztof 205  
Grabowski, Szymon 470  
Grabska, Ewa 297, 763  
Grzechca, D. 794  
Grzegorek, Maria 76, 156  
Gudra, Tadeusz 629
- Hennebert, Jean 429  
Henzel, Norbert 532  
Hippe, Zdzisław S. 503, 802, 808  
Horáček, Ondřej 44  
Horoba, Krzysztof 532  
Hrzebień, Maciej 550
- Iñesta, José M. 405  
Ingold, Rolf 429  
Iwanowski, Marcin 305
- Jabłoński, Marian 297  
Jankowski, Norbert 205

- Jantos, P. 794  
 Jaszczak, P. 613  
 Jezewski, Michal 532  
 Joanis, Eric 437  
 Jozwik, Adam 778  
 Jurek, Janusz 282  
 Justo, Raquel 421
- Kaczmarek, Piotr 581  
 Kamenický, Jan 44  
 Kasiński, Andrzej 124, 132, 581, 816  
 Klukowski, Leszek 232  
 Kněžík, Jan 526  
 Komorowski, D. 558  
 Koprowski, Robert 11  
 Korbicz, Józef 550  
 Korytkowski, Marcin 319  
 Korzyńska, Anna 604  
 Kosła, Paweł 394  
 Kostka, Paweł 356, 852  
 Kozak, Karol 224  
 Kozak, Marta 224  
 Kraft, Marek 132  
 Kuczumska, Barbara 802, 808  
 Kuhn, Roland 437  
 Kulikowski, Juliusz L. 68, 510  
 Kuniszyk-Józkowiak, Wiesława 445, 771  
 Kurzynski, Marek 485, 687  
 Kwitt, Roland 35
- Lagodzinski, Przemysław 108  
 Larkin, Samuel 437  
 Leski, Jacek 532  
 Lewandowski, Andrzej 824  
 Lewandowski, Tomasz 637  
 Lisniewska-Machorowska, Barbara 645  
 López-Cózar, Ramón 747  
 Luchowski, Leszek 274, 645
- Maciejewski, Henryk 831  
 Mahdian, Babak 187  
 Marnik, Joanna 454  
 Martínez-Trinidad, J. Francisco 27  
 Mashtalir, Volodymir 348  
 Maszewski, Mirosław 462  
 Maźbic-Kulma, Barbara 250  
 Mazurek, U. 852  
 Mazurkiewicz, Przemysław 589  
 Mazzariello, Claudio 705
- Meier, Hans-Günter 216  
 Meynet, Julien 663  
 Miciak, Mirosław 462  
 Migda, Joachim 629  
 Mokrzycki, Wojciech S. 100, 171  
 Mozerov, Mikhail 19, 164
- Nieczkowski, Tomasz 566  
 Noris, Basilio 663
- Obuchowicz, Andrzej 550, 566  
 Olvera-López, J. Arturo 27  
 Opielinski, Krzysztof J. 629
- Paszkowski, Bartosz 470  
 Paszyńska, Anna 297  
 Pedersoli, Marco 739  
 Penar, Wojciech 371  
 Piątek, Lukasz 503  
 Piecha, Jan 653  
 Pietka, D. 613  
 Piętka, Ewa 140, 671, 679  
 Płaczek, Bartłomiej 844  
 Porwik, Piotr 786  
 Potrzebowski, Henryk 386  
 Proença, Hugo 731  
 Przybyła, Tomasz 60  
 Przytułska, Małgorzata 510  
 Puchala, Dariusz 378
- Ramón Rico-Juan, Juan 405  
 Raniszewski, Marcin 258  
 Roca, Xavi 179  
 Rutkowski, J. 794  
 Rutkowski, Leszek 319
- Sadeghi, Mohammad T. 755  
 Saic, Stanislav 187  
 Salamończyk, Andrzej 100, 171  
 Sansone, Carlo 705  
 Sas, Jerzy 477, 485  
 Scherer, Rafał 319  
 Schmidt, Adam 124, 816  
 Sęp, Krzysztof 250, 386  
 Simiński, Krzysztof 493  
 Skabek, Krzysztof 196  
 Skomorowski, Marek 313  
 Skublewska-Paszkowska, Maria 84  
 Ślusarczyk, Grażyna 763  
 Smiatacz, Maciej 52



- Smolka, Bogdan 108  
Smolka, Elżbieta 445, 771  
Smolka, Jakub 84  
Sokolowska, Beata 778  
Spinczyk, Dominik 671  
Stańczak, Jarosław 386  
Staniek, Marcin 844  
Stapor, Katarzyna 224  
Straszecka, Ewa 542  
Strug, Barbara 266  
Stuhlsatz, André 216  
Suszyński, Waldemar 445  
Szczepaniak, Piotr S. 327  
Szczurowska, Izabela 771  
Szlachetko, Bogusław 824  
  
Tabatabaei, Mohammad S. 755  
Tarnawski, Michał 645  
Thiran, Jean-Philippe 663  
Tkacz, Ewaryst 356, 852  
Tomaka, Agnieszka 196, 645  
Tomczyk, Arkadiusz 148  
  
Topczewska, Magdalena 289, 348  
Torres, M. Inés 421  
  
Uhl, Andreas 35  
  
Villanueva, Juan J. 739  
  
Wendemuth, Andreas 216  
Wieclaw, Lukasz 786  
Wieclawek, A. 852  
Więclawek, Wojciech 140  
Wierzbicka, Diana 510  
Wiśniewski, Marek 445  
Wolczowski, Andrzej 687  
Wozniak, Michał 371  
Wrobel, Janusz 532  
Wrobel, Zygmunt 11  
  
Yatsymirskyy, Mykhaylo 378  
  
Zarychta, Piotr 695  
Zolnierek, Andrzej 477