

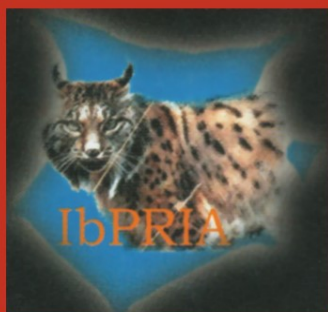
Jorge S. Marques  
Nicolás Pérez de la Blanca  
Pedro Pina (Eds.)

LNC3 3522

# Pattern Recognition and Image Analysis

Second Iberian Conference, IbPRIA 2005  
Estoril, Portugal, June 2005  
Proceedings, Part I

**I**  
Part I



 Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*University of Dortmund, Germany*

Madhu Sudan

*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos

*New York University, NY, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Moshe Y. Vardi

*Rice University, Houston, TX, USA*

Gerhard Weikum

*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

Jorge S. Marques   Nicolás Pérez de la Blanca  
Pedro Pina (Eds.)

# Pattern Recognition and Image Analysis

Second Iberian Conference, IbPRIA 2005  
Estoril, Portugal, June 7-9, 2005  
Proceedings, Part I

## Volume Editors

Jorge S. Marques

Instituto Superior Técnico, ISR

Torre Norte, Av. Rovisco Pais, 1049-001, Lisboa, Portugal

E-mail: jsm@isr.ist.utl.pt

Nicolás Pérez de la Blanca

Universidad de Granada, ETSI Informática

Departamento de Ciencias de la Computación e Inteligencia Artificial

Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain

E-mail: nicolas@ugr.es

Pedro Pina

Instituto Superior Técnico, CVRM

Av. Rovisco Pais, 1049-001 Lisboa, Portugal

E-mail: ppina@alfa.ist.utl.pt

Library of Congress Control Number: 2005926832

CR Subject Classification (1998): I.4, I.5, I.7, I.2.7, I.2.10

ISSN 0302-9743

ISBN-10 3-540-26153-2 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-26153-7 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, re-use of illustrations, recitation, broadcasting, reproduction on microfilms 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 to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik

Printed on acid-free paper SPIN: 11492429 06/3142 5 4 3 2 1 0

# Preface

IbPRIA 2005 (Iberian Conference on Pattern Recognition and Image Analysis) was the second of a series of conferences jointly organized every two years by the Portuguese and Spanish Associations for Pattern Recognition (APRP, AERFAI), with the support of the International Association for Pattern Recognition (IAPR).

This year, IbPRIA was hosted by the Institute for Systems and Robotics and the Geo-systems Center of the Instituto Superior Técnico and it was held in Estoril, Portugal. It provided the opportunity to bring together researchers from all over the world to discuss some of the most recent advances in pattern recognition and all areas of video, image and signal processing.

There was a very positive response to the Call for Papers for IbPRIA 2005. We received 292 full papers from 38 countries and 170 were accepted for presentation at the conference. The high quality of the scientific program of IbPRIA 2005 was due first to the authors who submitted excellent contributions and second to the dedicated collaboration of the international Program Committee and the other researchers who reviewed the papers. Each paper was reviewed by two reviewers, in a blind process. We would like to thank all the authors for submitting their contributions and for sharing their research activities. We are particularly indebted to the Program Committee members and to all the reviewers for their precious evaluations, which permitted us to set up this publication.

We were also very pleased to benefit from the participation of the invited speakers Prof. David Lowe, University of British Columbia (Canada), Prof. Wiro Niessen, University of Utrecht (The Netherlands) and Prof. Isidore Rigoutsos, IBM Watson Research Center (USA). We would like to express our sincere gratitude to these world-renowned experts.

We would like to thank Prof. João Sanches and Prof. João Paulo Costeira of the Organizing Committee, in particular for the management of the Web page and the submission system software.

Finally, we were very pleased to welcome all the participants who attended IbPRIA 2005. We are looking forward to meeting you at the next edition of IbPRIA, in Spain in 2007.

Estoril, June 2005

Jorge S. Marques  
Nicolás Pérez de la Blanca  
Pedro Pina

## **Conference Chairs**

Jorge S. Marques  
Nicolás Pérez de la Blanca  
Pedro Pina

Instituto Superior Técnico  
University of Granada  
Instituto Superior Técnico

## **Organizing Committee**

João M. Sanches  
João Paulo Costeira

Instituto Superior Técnico  
Instituto Superior Técnico

## **Invited Speakers**

David Lowe  
Wiro Niessen  
Isidore Rigoutsos

University of British Columbia, Canada  
University of Utrecht, The Netherlands  
IBM Watson Research Center, USA

## **Supported by**

Fundação Oriente, Lisbon  
Fundação para a Ciência e Tecnologia  
HP Portugal  
Institute for Systems and Robotics, Lisbon  
International Association for Pattern Recognition

**Program Committee**

Jake Aggarwal	University of Texas, USA
Hélder Araújo	University of Coimbra, Portugal
José Benedi	Polytechnic University of Valencia, Spain
Isabelle Bloch	ENST, France
Hervé Boulard	EPFL, Switzerland
Patrick Bouthemy	IRISA, France
Horst Bunke	University of Bern, Switzerland
Aurélio Campilho	University of Porto, Portugal
Gilles Celeux	Université Paris-Sud, France
Luigi Cordella	University of Naples, Italy
Alberto Del Bimbo	University of Florence, Italy
Hervé Delinguette	INRIA, France
Rachid Deriche	INRIA, France
José Dias	Instituto Superior Técnico, Portugal
Robert Duin	University of Delft, The Netherlands
Mário Figueiredo	Instituto Superior Técnico, Portugal
Ana Fred	Instituto Superior Técnico, Portugal
Andrew Gee	University of Cambridge, UK
Mohamed Kamel	University of Waterloo, Canada
Aggelos Katsaggelos	Northwestern University, USA
Joseph Kittler	University of Surrey, UK
Seong-Whan Lee	University of Korea, Korea
Ana Mendonça	University of Porto, Portugal
Hermann Ney	University of Aachen, Germany
Wiro Niessen	University of Utrecht, The Netherlands
Francisco Perales	Universitat de les Illes Balears, Spain
Maria Petrou	University of Surrey, UK
Armando Pinho	University of Aveiro, Portugal
Ioannis Pitas	University of Thessaloniki, Greece
Filiberto Pla	University Jaume I, Spain
Richard Prager	University of Cambridge, UK
José Principe	University of Florida, USA
Ian Reid	University of Oxford, UK
Gabriella Sanniti di Baja	Istituto di Cibernetica, Italy
Beatriz Santos	University of Aveiro, Portugal
José Santos-Victor	Instituto Superior Técnico, Portugal
Joan Serrat	Universitat Autònoma de Barcelona, Spain
Yoshiaki Shirai	Osaka University, Japan
Pierre Soille	Joint Research Centre, Italy
Karl Tombre	LORIA, France
M. Ines Torres	University of the Basque Country, Spain
Emanuele Trucco	Heriot-Watt University, UK
Alessandro Verri	University of Genoa, Italy
Max Viergever	University of Utrecht, The Netherlands
Joachim Weickert	Saarland University, Germany

**Reviewers**

Arnaldo Abrantes  
Luís Alexandre  
René Alquézar  
Juan Carlos Amengual  
Teresa Barata  
Jorge Barbosa  
Jorge Batista  
Luis Baumela  
Alexandre Bernardino  
Javier Binefa  
Hans Du Buf  
Francisco Casacuberta  
Miguel Velhote Correia  
Paulo Correia  
João P. Costeira  
Jose Manuel Fuertes  
José Gaspar  
Edwin Hancock  
Francisco Mario Hernández  
Arturo De La Escalera Hueso  
Jose Manuel Iñesta  
Alfons Juan  
João Miranda Lemos  
Manuel Lucena Lopez  
Javier Lorenzo  
Maria Angeles Lopez Malo  
Elisa Martínez Marroquín  
Jesus Chamorro Martinez  
Eduard Montseny Masip  
Nicolás Guil Mata  
Luisa Micó  
Rafael Molina

Ramón A. Mollineda  
Jacinto Nascimento  
Jesus Ariel Carrasco Ochoa  
Paulo Oliveira  
António Guedes Oliveira  
Arlindo Oliveira  
Antonio Adan Oliver  
José Oncina  
Roberto Paredes  
Antonio Miguel Peinado  
Fernando Pereira  
André Puga  
Francesc Josep Ferri Rabasa  
Juan Mendez Rodriguez  
Antoni Grau Saldes  
João M. Sanches  
José Salvador Sánchez  
Modesto Castrillon Santana  
José Ruiz Shulcloper  
Jorge Alves Silva  
Margarida Silveira  
António Jorge Sousa  
João M. Sousa  
João Tavares  
António J.S. Teixeira  
Ana Maria Tomé  
Jose Ramon Fernandez Vidal  
Enrique Vidal  
Juan Jose Villanueva  
Jordi Vitrià



# Table of Contents, Part I

---

## I Computer Vision

---

An Invariant and Compact Representation for Unrestricted Pose Estimation . . . . .	3
<i>Robert Söderberg, Klas Nordberg, and Gösta Granlund</i>	
Gabor Parameter Selection for Local Feature Detection . . . . .	11
<i>Plinio Moreno, Alexandre Bernardino, and José Santos-Victor</i>	
Real-Time Tracking Using Multiple Target Models . . . . .	20
<i>Manuel J. Lucena, José M. Fuertes, and Nicolás Pérez de la Blanca</i>	
Efficient Object-Class Recognition by Boosting Contextual Information . . . . .	28
<i>Jaume Amores, Nicu Sebe, and Petia Radeva</i>	
Illumination Intensity, Object Geometry and Highlights Invariance in Multispectral Imaging . . . . .	36
<i>Raúl Montoliu, Filiberto Pla, and Arnoud C. Klaren</i>	
Local Single-Patch Features for Pose Estimation Using the Log-Polar Transform .	44
<i>Fredrik Viksten and Anders Moe</i>	
Dealing with Multiple Motions in Optical Flow Estimation . . . . .	52
<i>Jesús Chamorro-Martínez, Javier Martínez-Baena, Elena Galán-Perales, and Beén Prados-Suárez</i>	
Conversion into Three-Dimensional Implicit Surface Representation from <i>Topological Active Volumes</i> Based Segmentation . . . . .	60
<i>José Rouco, Noelia Barreira, Manuel G. Penedo, and Xosé M. Pardo</i>	
Automatic Matching and Motion Estimation from Two Views of a Multiplane Scene . . . . .	69
<i>Gonzalo López-Nicolás, Carlos Sagüés, and José J. Guerrero</i>	
Contextual Soccer Detection Using Mosaicing Techniques . . . . .	77
<i>Lluís Barceló and Xavier Binefa</i>	
Probabilistic Image-Based Tracking: Improving Particle Filtering . . . . .	85
<i>Daniel Rowe, Ignasi Rius, Jordi González, Xavier Roca, and Juan J. Villanueva</i>	
A Framework to Integrate Particle Filters for Robust Tracking in Non-stationary Environments . . . . .	93
<i>Francesc Moreno-Noguer and Alberto Sanfeliu</i>	

Stereo Reconstruction of a Submerged Scene . . . . .	102
<i>Ricardo Ferreira, João P. Costeira, and João A. Santos</i>	
A Functional Simplification of the BCS/FCS Image Segmentation . . . . .	110
<i>Pablo Martínez, Miguel Pinzolas, Juan López Coronado, and Daniel García</i>	
From Moving Edges to Moving Regions . . . . .	119
<i>Loic Biancardini, Eva Dokladalova, Serge Beucher, and Laurent Letellier</i>	
Polygon Optimisation for the Modelling of Planar Range Data . . . . .	128
<i>Samuel Nunes, Daniel Almeida, Eddy Loke, and Hans du Buf</i>	
Stereo Vision System with the Grouping Process of Multiple Reaction-Diffusion Models . . . . .	137
<i>Atsushi Nomura, Makoto Ichikawa, and Hidetoshi Miike</i>	
Registration of Moving Surfaces by Means of One-Shot Laser Projection . . . . .	145
<i>Carles Matabosch, David Fofi, Joaquim Salvi, and Josep Forest</i>	
A Computer Vision Sensor for Panoramic Depth Perception . . . . .	153
<i>Radu Orghidan, El Mustapha Mouaddib, and Joaquim Salvi</i>	
Probabilistic Object Tracking Based on Machine Learning and Importance Sampling . . . . .	161
<i>Peihua Li and Haijing Wang</i>	
A Calibration Algorithm for POX-Slits Camera . . . . .	168
<i>Nuno Martins and Hélder Araújo</i>	
Vision-Based Interface for Integrated Home Entertainment System . . . . .	176
<i>Jae Sik Chang, Sang Ho Kim, and Hang Joon Kim</i>	
A Proposal for a Homeostasis Based Adaptive Vision System . . . . .	184
<i>Javier Lorenzo-Navarro, Daniel Hernández, Cayetano Guerra, and José Isern-González</i>	
Relaxed Grey-World: Computational Colour Constancy by Surface Matching . . . .	192
<i>Francesc Tous, María Vanrell, and Ramón Baldrich</i>	
A Real-Time Driver Visual Attention Monitoring System . . . . .	200
<i>Jorge P. Batista</i>	
An Approach to Vision-Based Person Detection in Robotic Applications . . . . .	209
<i>Carlos Castillo and Carolina Chang</i>	
A New Approach to the Template Update Problem . . . . .	217
<i>Cayetano Guerra, Mario Hernández, Antonio Domínguez, and Daniel Hernández</i>	

---

## II Shape and Matching

---

Contour-Based Image Registration Using Mutual Information . . . . .	227
<i>Nancy A. Álvarez, José M. Sanchiz, Jorge Badenas, Filiberto Pla, and Gustavo Casañ</i>	
Improving Correspondence Matching Using Label Consistency Constraints . . . . .	235
<i>Hongfang Wang and Edwin R. Hancock</i>	
The Euclidean Distance Transform Applied to the FCC and BCC Grids . . . . .	243
<i>Robin Strand</i>	
Matching Deformable Regions Using Local Histograms of Differential Invariants . . . . .	251
<i>Nicolás Pérez de la Blanca, José M. Fuertes, and Manuel J. Lucena</i>	
A Global-to-Local Matching Strategy for Registering Retinal Fundus Images . . . . .	259
<i>Xinge You, Bin Fang, Zhenyu He, and Yuan Yan Tang</i>	
A Model-Based Method for Face Shape Recovery . . . . .	268
<i>William A.P. Smith and Edwin R. Hancock</i>	
Visual Detection of Hexagonal Headed Bolts Using Method of Frames and Matching Pursuit . . . . .	277
<i>Pier Luigi Mazzeo, Ettore Stella, Nicola Ancona, and Arcangelo Distante</i>	
A New Region-Based Active Contour for Object Extraction Using Level Set Method . . . . .	285
<i>Lishui Cheng, Jie Yang, and Xian Fan</i>	
Improving ASM Search Using Mixture Models for Grey-Level Profiles . . . . .	292
<i>Yanong Zhu, Mark Fisher, and Reyer Zwiggelaar</i>	
Human Figure Segmentation Using Independent Component Analysis . . . . .	300
<i>Grégory Rogez, Carlos Orrite-Uruñuela, and Jesús Martínez-del-Rincón</i>	
Adaptive Window Growing Technique for Efficient Image Matching . . . . .	308
<i>Bogusław Cyganek</i>	
Managing Resolution in Digital Elevation Models Using Image Processing Techniques . . . . .	316
<i>Rolando Quintero, Serguei Levachkine, Miguel Torres, Marco Moreno, and Giovanni Guzman</i>	
Object Image Retrieval by Shape Content in Complex Scenes Using Geometric Constraints . . . . .	325
<i>Agnés Borràs and Josep Lladós</i>	

---

### III Image and Video Processing

---

A Real-Time Gabor Primal Sketch for Visual Attention . . . . . 335  
*Alexandre Bernardino and José Santos-Victor*

Bayesian Reconstruction of Color Images Acquired with a Single CCD . . . . . 343  
*Miguel Vega, Rafael Molina, and Aggelos K. Katsaggelos*

A Fast and Exact Algorithm for Total Variation Minimization . . . . . 351  
*Jérôme Darbon and Marc Sigelle*

Phase Unwrapping via Graph Cuts . . . . . 360  
*José M. Bioucas-Dias and Gonçalo Valadão*

A New Fuzzy Multi-channel Filter for the Reduction of Impulse Noise . . . . . 368  
*Stefan Schulte, Valérie De Witte, Mike Nachtegael,  
 Dietrich Van der Weken, and Etienne E. Kerre*

Enhancement and Cleaning of Handwritten Data by Using Neural Networks . . . . . 376  
*José Luis Hidalgo, Salvador España, María José Castro,  
 and José Alberto Pérez*

Zerotree Wavelet Based Image Quilting for Fast Texture Synthesis . . . . . 384  
*Dhammike S. Wickramanayake, Eran A. Edirisinghe, and Helmut E. Bez*

Semantic Feature Extraction Based on Video Abstraction  
 and Temporal Modeling . . . . . 392  
*Kisung Lee*

Video Retrieval Using an EDL-Based Timeline . . . . . 401  
*José San Pedro, Nicolas Denis, and Sergio Domínguez*

---

### IV Image and Video Coding

---

A New Secret Sharing Scheme for Images  
 Based on Additive 2-Dimensional Cellular Automata . . . . . 411  
*Gonzalo Álvarez Marañón, Luis Hernández Encinas,  
 and Ángel Martín del Rey*

A Fast Motion Estimation Algorithm  
 Based on Diamond and Triangle Search Patterns . . . . . 419  
*Yun Cheng, Zhiying Wang, Kui Dai, and Jianjun Guo*

A Watermarking Scheme Based on Discrete Non-separable Wavelet Transform . . 427  
*Jianwei Yang, Xinge You, Yuan Yan Tang, and Bin Fang*

A Fast Run-Length Algorithm for Wavelet Image Coding with Reduced Memory Usage . . . . .	435
<i>Jose Oliver and Manuel P. Malumbres</i>	

---

## V Face Recognition

---

Multiple Face Detection at Different Resolutions for Perceptual User Interfaces . .	445
<i>Modesto Castrillón-Santana, Javier Lorenzo-Navarro, Oscar Déniz-Suárez, José Isern-González, and Antonio Falcón-Martel</i>	
Removing Shadows from Face Images Using ICA . . . . .	453
<i>Jun Liu, Xiangsheng Huang, and Yangsheng Wang</i>	
An Analysis of Facial Description in Static Images and Video Streams . . . . .	461
<i>Modesto Castrillón-Santana, Javier Lorenzo-Navarro, Daniel Hernández-Sosa, and Yeray Rodríguez-Domínguez</i>	
Recognition of Facial Gestures Based on Support Vector Machines . . . . .	469
<i>Attila Fazekas and István Sánta</i>	
Performance Driven Facial Animation by Appearance Based Tracking . . . . .	476
<i>José Miguel Buenaposada, Enrique Muñoz, and Luis Baumela</i>	
Color Distribution Tracking for Facial Analysis . . . . .	484
<i>Juan José Gracia-Roche, Carlos Orrite, Emiliano Bernués, and José Elías Herrero</i>	
Head Gesture Recognition Based on Bayesian Network . . . . .	492
<i>Peng Lu, Xiangsheng Huang, Xinshan Zhu, and Yangsheng Wang</i>	
Detection and Tracking of Face by a Walking Robot . . . . .	500
<i>Do Joon Jung, Chang Woo Lee, and Hang Joon Kim</i>	

---

## VI Human Activity Analysis

---

Appearance-Based Recognition of Words in American Sign Language . . . . .	511
<i>Morteza Zahedi, Daniel Keysers, and Hermann Ney</i>	
Robust Person-Independent Visual Sign Language Recognition . . . . .	520
<i>Jörg Zieren and Karl-Friedrich Kraiss</i>	
A 3D Dynamic Model of Human Actions for Probabilistic Image Tracking . . . . .	529
<i>Ignasi Rius, Daniel Rowe, Jordi González, and Xavier Roca</i>	
Extracting Motion Features for Visual Human Activity Representation . . . . .	537
<i>Filiberto Pla, Pedro Ribeiro, José Santos-Victor, and Alexandre Bernardino</i>	

Modelling Spatial Correlation and Image Statistics  
for Improved Tracking of Human Gestures . . . . . 545  
*Rik Bellens, Sidharta Gautama, and Johan D'Haeyer*

Fast and Accurate Hand Pose Detection for Human-Robot Interaction . . . . . 553  
*Luis Antón-Canalís, Elena Sánchez-Nielsen,  
and Modesto Castrillón-Santana*

---

## VII Surveillance

---

Performance Analysis of Homomorphic Systems for Image Change Detection . . . 563  
*Gonzalo Pajares, José Jaime Ruz, and Jesús Manuel de la Cruz*

Car License Plates Extraction and Recognition  
Based on Connected Components Analysis and HMM Decoding . . . . . 571  
*David Llorens, Andrés Marzal, Vicente Palazón, and Juan M. Vilar*

Multi-resolution Image Analysis for Vehicle Detection . . . . . 579  
*Cristina Hilario, Juan Manuel Collado, José Maria Armingol,  
and Arturo de la Escalera*

A Novel Adaptive Gaussian Mixture Model for Background Subtraction . . . . . 587  
*Jian Cheng, Jie Yang, and Yue Zhou*

Intelligent Target Recognition Based on Wavelet Adaptive Network  
Based Fuzzy Inference System . . . . . 594  
*Engin Avci, Ibrahim Turkoglu, and Mustafa Poyraz*

---

## VIII Robotics

---

HMM-Based Gesture Recognition for Robot Control . . . . . 607  
*Hye Sun Park, Eun Yi Kim, Sang Su Jang, Se Hyun Park,  
Min Ho Park, and Hang Joon Kim*

PCA Positioning Sensor Characterization for Terrain Based Navigation of UVs . . 615  
*Paulo Oliveira*

Monte Carlo Localization Using SIFT Features . . . . . 623  
*Arturo Gil, Óscar Reinoso, Asunción Vicente,  
César Fernández, and Luis Payá*

A New Method for the Estimation of the Image Jacobian  
for the Control of an Uncalibrated Joint System . . . . . 631  
*Jose M. Sebastián, Lizardo Pari, Carolina González, and Luis Ángel*

Accelerometer Based Gesture Recognition Using Continuous HMMs . . . . . 639  
*Timo Pylvänäinen*

An Approach to Improve Online Hand-Eye Calibration . . . . .	647
<i>Fanhuai Shi, Jianhua Wang, and Yuncai Liu</i>	

---

## **IX Hardware Architectures**

---

Image Processing Application Development: From Rapid Prototyping to SW/HW Co-simulation and Automated Code Generation . . . . .	659
<i>Cristina Vicente-Chicote, Ana Toledo, and Pedro Sánchez-Palma</i>	
Xilinx System Generator Based HW Components for Rapid Prototyping of Computer Vision SW/HW Systems . . . . .	667
<i>Ana Toledo, Cristina Vicente-Chicote, Juan Suardíaz, and Sergio Cuenca</i>	
2-D Discrete Cosine Transform (DCT) on Meshes with Hierarchical Control Modes . . . . .	675
<i>Cheong-Ghil Kim, Su-Jin Lee, and Shin-Dug Kim</i>	
Domain-Specific Codesign for Automated Visual Inspection Systems . . . . .	683
<i>Sergio Cuenca, Antonio Cámara, Juan Suardíaz, and Ana Toledo</i>	
Hardware-Accelerated Template Matching . . . . .	691
<i>Raúl Cabido, Antonio S. Montemayor, and Ángel Sánchez</i>	
<b>Author Index</b> . . . . .	699

# Table of Contents, Part II

---

## I Statistical Pattern Recognition

---

Testing Some Improvements of the Fukunaga and Narendra's Fast Nearest Neighbour Search Algorithm in a Spelling Task . . . . .	3
<i>Eva Gómez-Ballester, Luisa Micó, and Jose Oncina</i>	
Solving Particularization with Supervised Clustering Competition Scheme . . . . .	11
<i>Oriol Pujol and Petia Radeva</i>	
Adaptive Optimization with Constraints: Convergence and Oscillatory Behaviour . . . . .	19
<i>Fernando J. Coito and João M. Lemos</i>	
Data Characterization for Effective Prototype Selection . . . . .	27
<i>Ramón A. Mollineda, J. Salvador Sánchez, and José M. Sotoca</i>	
A Stochastic Approach to Wilson's Editing Algorithm . . . . .	35
<i>Fernando Vázquez, J. Salvador Sánchez, and Filiberto Pla</i>	
Parallel Perceptrons, Activation Margins and Imbalanced Training Set Pruning . . . . .	43
<i>Iván Cantador and José R. Dorronsoro</i>	
Boosting Statistical Local Feature Based Classifiers for Face Recognition . . . . .	51
<i>Xiangsheng Huang and Yangsheng Wang</i>	
Dynamic and Static Weighting in Classifier Fusion . . . . .	59
<i>Rosa M. Valdovinos, J. Salvador Sánchez, and Ricardo Barandela</i>	
A Novel One-Parameter Regularized Kernel Fisher Discriminant Method for Face Recognition . . . . .	67
<i>Wensheng Chen, Pongchi Yuen, Jian Huang, and Daoqing Dai</i>	
AutoAssign – An Automatic Assignment Tool for Independent Components . . . . .	75
<i>Matthias Böhm, Kurt Stadlthanner, Ana M. Tomé, Peter Gruber, Ana R. Teixeira, Fabian J. Theis, Carlos G. Puntonet, and Elmar W. Lang</i>	
Improving the Discrimination Capability with an Adaptive Synthetic Discriminant Function Filter . . . . .	83
<i>J. Ángel González-Fraga, Víctor H. Díaz-Ramírez, Vitaly Kober, and Josué Álvarez-Borrego</i>	



Globally Exponential Stability of Non-autonomous Delayed Neural Networks . . . 91  
*Qiang Zhang, Wenbing Liu, Xiaopeng Wei, and Jin Xu*

---

## II Syntactical Pattern Recognition

---

Comparison of Two Different Prediction Schemes for the Analysis  
of Time Series of Graphs . . . . . 99  
*Horst Bunke, Peter Dickinson, and Miro Kraetzl*

Grouping of Non-connected Structures by an Irregular Graph Pyramid . . . . . 107  
*Walter G. Kropatsch and Yll Haxhimusa*

An Adjacency Grammar to Recognize Symbols and Gestures  
in a Digital Pen Framework . . . . . 115  
*Joan Mas, Gemma Sánchez, and Josep Lladós*

Graph Clustering Using Heat Content Invariants . . . . . 123  
*Bai Xiao and Edwin R. Hancock*

Matching Attributed Graphs: 2nd-Order Probabilities  
for Pruning the Search Tree . . . . . 131  
*Francesc Serratos and Alberto Sanfeliu*

Synthesis of Median Spectral Graph . . . . . 139  
*Miquel Ferrer, Francesc Serratos, and Alberto Sanfeliu*

Feature Selection for Graph-Based Image Classifiers . . . . . 147  
*Bertrand Le Saux and Horst Bunke*

Machine Learning with Seriated Graphs . . . . . 155  
*Hang Yu and Edwin R. Hancock*

Time Reduction of Stochastic Parsing with Stochastic Context-Free Grammars . . 163  
*Joan Andreu Sánchez and José Miguel Benedí*

---

## III Image Analysis

---

Segment Extraction Using Burns Principles  
in a Pseudo-color Fuzzy Hough Transform . . . . . 175  
*Marta Penas, María J. Carreira, Manuel G. Penedo, and Cástor Mariño*

Texture Interpolation Using Ordinary Kriging . . . . . 183  
*Sunil Chandra, Maria Petrou, and Roberta Piroddi*

Spectral Methods in Image Segmentation: A Combined Approach . . . . . 191  
*Fernando C. Monteiro and Aurélio C. Campilho*

Mathematical Morphology in Polar-Logarithmic Coordinates. Application to Erythrocyte Shape Analysis . . . . .	199
<i>Miguel A. Luengo-Oroz, Jesús Angulo, Georges Flandrin, and Jacques Klossa</i>	
Signal Subspace Identification in Hyperspectral Linear Mixtures . . . . .	207
<i>José M.P. Nascimento and José M.B. Dias</i>	
Automatic Selection of Multiple Texture Feature Extraction Methods for Texture Pattern Classification . . . . .	215
<i>Domènec Puig and Miguel Ángel Garcia</i>	
Dynamic Texture Recognition Using Normal Flow and Texture Regularity . . . . .	223
<i>Renaud Péteri and Dmitry Chetverikov</i>	
Detector of Image Orientation Based on Borda-Count. . . . .	231
<i>Loris Nanni and Alessandra Lumini</i>	
Color Image Segmentation Using Acceptable Histogram Segmentation . . . . .	239
<i>Julie Delon, Agnes Desolneux, Jose Luis Lisani, and Ana Belen Petro</i>	
Adding Subsurface Attenuation to the Beckmann-Kirchhoff Theory . . . . .	247
<i>Hossein Ragheb and Edwin R. Hancock</i>	
Multi-scale Cortical Keypoint Representation for Attention and Object Detection . . . . .	255
<i>João Rodrigues and Hans du Buf</i>	
Evaluation of Distances Between Color Image Segmentations . . . . .	263
<i>Jaume Vergés-Llahí and Alberto Sanfeliu</i>	
An Algorithm for the Detection of Multiple Concentric Circles . . . . .	271
<i>Margarida Silveira</i>	
Image Corner Detection Using Hough Transform . . . . .	279
<i>Sung Kwan Kang, Young Chul Choung, and Jong An Park</i>	
Dissimilarity Measures for Visual Pattern Partitioning . . . . .	287
<i>Raquel Dosil, Xosé R. Fdez-Vidal, and Xosé M. Pardo</i>	
A Comparative Study of Highlights Detection and Elimination by Color Morphology and Polar Color Models . . . . .	295
<i>Francisco Ortiz, Fernando Torres, and Pablo Gil</i>	
Algorithm for Crest Detection Based on Graph Contraction . . . . .	303
<i>Nazha Selmaoui</i>	
A Learning Framework for Object Recognition on Image Understanding . . . . .	311
<i>Xavier Muñoz, Anna Bosch, Joan Martí, and Joan Espunya</i>	

A Roof Edge Detection Model . . . . . 319  
*Qing H. Zhang, Song Gao, and Tien D. Bui*

A Dynamic Stroke Segmentation Technique for Sketched Symbol Recognition . . . 328  
*Vincenzo Deufemia and Michele Risi*

Application of Wavelet Transforms and Bayes Classifier  
to Segmentation of Ultrasound Images . . . . . 336  
*Pawel Kieś*

Use of Neural Networks in Automatic Caricature Generation:  
An Approach Based on Drawing Style Capture . . . . . 343  
*Rupesh N. Shet, Ka H. Lai, Eran A. Edirisinghe, and Paul W.H. Chung*

---

## IV Document Analysis

---

Information Theoretic Text Classification Using the Ziv-Merhav Method . . . . . 355  
*David Pereira Coutinho and Mário A.T. Figueiredo*

Spontaneous Handwriting Text Recognition and Classification  
Using Finite-State Models . . . . . 363  
*Alejandro Héctor Toselli, Moisés Pastor, Alfons Juan, and Enrique Vidal*

Combining Global and Local Threshold to Binarize Document of Images . . . . . 371  
*Elise Gabarra and Antoine Tabbone*

Extended Bi-gram Features in Text Categorization . . . . . 379  
*Xian Zhang and Xiaoyan Zhu*

Combining Fuzzy Clustering and Morphological Methods  
for Old Documents Recovery . . . . . 387  
*João R. Caldas Pinto, Lourenço Bandeira, João M.C. Sousa, and Pedro Pina*

---

## V Bioinformatics

---

A New Algorithm for Pattern Optimization  
in Protein-Protein Interaction Extraction System . . . . . 397  
*Yu Hao, Xiaoyan Zhu, and Ming Li*

Curvature Based Clustering for DNA Microarray Data Analysis . . . . . 405  
*Emil Saucan and Eli Appleboim*

Support Vector Machines for HIV-1 Protease Cleavage Site Prediction . . . . . 413  
*Loris Nanni and Alessandra Lumini*

Medial Grey-Level Based Representation for Proteins in Volume Images . . . . . 421  
*Ida-Maria Sintorn, Magnus Gedda, Susana Mata, and Stina Svensson*

---

## VI Medical Imaging

---

Automatic Classification of Breast Tissue . . . . .	431
<i>Arnau Oliver, Jordi Freixenet, Anna Bosch, David Raba, and Reyer Zwiggelaar</i>	
On Reproducibility of Ultrasound Image Classification . . . . .	439
<i>Martin Švec, Radim Šára, and Daniel Smutek</i>	
Prior Based Cardiac Valve Segmentation in Echocardiographic Sequences: Geodesic Active Contour Guided by Region and Shape Prior . . . . .	447
<i>Yanfeng Shang, Xin Yang, Ming Zhu, Biao Jin, and Ming Liu</i>	
Bayesian Reconstruction for Transmission Tomography with Scale Hyperparameter Estimation . . . . .	455
<i>Antonio López, Rafael Molina, and Aggelos K. Katsaggelos</i>	
Automatic Segmentation and Registration of Lung Surfaces in Temporal Chest CT Scans . . . . .	463
<i>Helen Hong, Jeongjin Lee, Yeni Yim, and Yeong Gil Shin</i>	
Breast Segmentation with Pectoral Muscle Suppression on Digital Mammograms . . . . .	471
<i>David Raba, Arnau Oliver, Joan Martí, Marta Peracaula, and Joan Espunya</i>	
Semivariogram and SGLDM Methods Comparison for the Diagnosis of Solitary Lung Nodule . . . . .	479
<i>Aristófanés C. Silva, Anselmo C. Paiva, Paulo C.P. Carvalho, and Marcelo Gattass</i>	
Anisotropic 3D Reconstruction and Restoration for Rotation-Scanning 4D Echocardiographic Images Based on MAP-MRF . . . . .	487
<i>Qiang Guo, Xin Yang, Ming Zhu, and Kun Sun</i>	
Comparison of Feature Extraction Methods for Breast Cancer Detection . . . . .	495
<i>Rafael Llobet, Roberto Paredes, and Juan C. Pérez-Cortés</i>	

---

## VII Biometrics

---

Multiscale Approach for Thinning Ridges of Fingerprint . . . . .	505
<i>Xinge You, Bin Fang, Yuan Yan Tang, and Jian Huang</i>	
Discriminative Face Recognition Through Gabor Responses and Sketch Distortion . . . . .	513
<i>Daniel González-Jiménez and José Luis Alba-Castro</i>	

Compact and Robust Fingerprints Using DCT Coefficients of Key Blocks . . . . . 521  
*Sheng Tang, Jin Tao Li, and Yong Dong Zhang*

Fingerprint Matching Using Minutiae Coordinate Systems . . . . . 529  
*Farid Benhammadi, Hamid Hentous, Kadda Bey-Beghdad,  
and Mohamed Aissani*

The Contribution of External Features to Face Recognition . . . . . 537  
*Ágata Lapedriza, David Masip, and Jordi Vitrià*

Iris Recognition Based on Quadratic Spline Wavelet Multi-scale Decomposition . 545  
*Xing Ming, Xiaodong Zhu, and Zhengxuan Wang*

---

## VIII Speech Recognition

---

An Utterance Verification Algorithm in Keyword Spotting System . . . . . 555  
*Haisheng Dai, Xiaoyan Zhu, Yupin Luo, and Shiyuan Yang*

A Clustering Algorithm for the Fast Match of Acoustic Conditions  
in Continuous Speech Recognition . . . . . 562  
*Luis Javier Rodríguez and M. Inés Torres*

Adaptive Signal Models for Wide-Band Speech and Audio Compression . . . . . 571  
*Pedro Vera-Candeas, Nicolás Ruiz-Reyes, Manuel Rosa-Zurera,  
Juan C. Cuevas-Martinez, and Francisco López-Ferreras*

Cryptographic-Speech-Key Generation Architecture Improvements . . . . . 579  
*L. Paola García-Perera, Juan A. Nolzaco-Flores, and Carlos Mex-Perera*

Performance of a SCFG-Based Language Model  
with Training Data Sets of Increasing Size . . . . . 586  
*Joan Andreu Sánchez, José Miguel Benedí, and Diego Linares*

Speaker Dependent ASRs for Huastec and Western-Huastec Náhuatl Languages . 595  
*Juan A. Nolzaco-Flores, Luis R. Salgado-Garza, and Marco Peña-Díaz*

---

## IX Natural Language Analysis

---

Phrase-Based Alignment Models for Statistical Machine Translation . . . . . 605  
*Jesús Tomás, Jaime Lloret, and Francisco Casacuberta*

Automatic Segmentation of Bilingual Corpora:  
A Comparison of Different Techniques . . . . . 614  
*Ismael García Varea, Daniel Ortiz, Francisco Nevado,  
Pedro A. Gómez, and Francisco Casacuberta*

Word Translation Disambiguation Using Multinomial Classifiers . . . . . 622  
*Jesús Andrés, José R. Navarro, Alfons Juan, and Francisco Casacuberta*

Different Approaches to Bilingual Text Classification Based on Grammatical Inference Techniques . . . . .	630
<i>Jorge Civera, Elsa Cubel, Alfons Juan, and Enrique Vidal</i>	
Semantic Similarity Between Sentences Through Approximate Tree Matching . . .	638
<i>Francisco Jose Ribadas, Manuel Vilares, and Jesus Vilares</i>	

---

## **X Applications**

---

A Text Categorization Approach for Music Style Recognition . . . . .	649
<i>Carlos Pérez-Sancho, José M. Iñesta, and Jorge Calera-Rubio</i>	
The MORFO3D Foot Database . . . . .	658
<i>José García-Hernández, Stella Heras, Alfons Juan, Roberto Paredes, Beatriz Nácher, Sandra Alemany, Enrique Alcántara, and Juan Carlos González</i>	
Fast Surface Grading Using Color Statistics in the CIE Lab Space . . . . .	666
<i>Fernando López, José Miguel Valiente, Ramón Baldrich, and María Vanrell</i>	
Quantitative Identification of Marbles Aesthetical Features . . . . .	674
<i>Roberto Bruno, Lorenza Cuoghi, and Pascal Laurengé</i>	
Leather Inspection Based on Wavelets . . . . .	682
<i>João Luís Sobral</i>	
Multispectral Image Segmentation by Energy Minimization for Fruit Quality Estimation . . . . .	689
<i>Adolfo Martínez-Usó, Filiberto Pla, and Pedro García-Sevilla</i>	
Thresholding Methods on MRI to Evaluate Intramuscular Fat Level on Iberian Ham . . . . .	697
<i>Mar Ávila, Marisa Luisa Durán, Andres Caro, Teresa Antequera, and Ramiro Gallardo</i>	
Automatic Mask Extraction for PIV-Based Dam-Break Analysis . . . . .	705
<i>Alberto Biancardi, Paolo Ghilardi, and Matteo Pagliardi</i>	
Analysis of Meso Textures of Geomaterials Through Haralick Parameters . . . . .	713
<i>Margarida Tabora Duarte and Joanne Mae Robison Fernlund</i>	
Decision Fusion for Target Detection Using Multi-spectral Image Sequences from Moving Cameras . . . . .	720
<i>Luis López-Gutiérrez and Leopoldo Altamirano-Robles</i>	
<b>Author Index</b> . . . . .	729

**Part I**

**Computer Vision**

# An Invariant and Compact Representation for Unrestricted Pose Estimation

Robert Söderberg, Klas Nordberg, and Gösta Granlund

Computer Vision Laboratory  
Department of Electrical Engineering  
Linköping University, SE-581 83 Linköping  
{soderberg, klas, gosta}@isy.liu.se

**Abstract.** This paper describes a novel compact representation of local features called the tensor doublet. The representation generates a four dimensional feature vector which is significantly less complex than other approaches, such as Lowe's 128 dimensional feature vector. Despite its low dimensionality, we demonstrate here that the tensor doublet can be used for pose estimation, where the system is trained for an object and evaluated on images with cluttered background and occlusion.

## 1 Introduction

Pose estimation of objects is of great interest in several industrial applications, especially in the unsolved bin picking problem. Industrial automation of today demands very dynamic automation systems since the geometry of the products changes faster than before. As a consequence, old systems where the objects are placed in fixtures, will not be sufficient in the future. Instead we need more advanced procedures that can find the pose of objects, guide the robot toward the objects and finally grasp them.

Over the years several algorithms have been developed for view centered pose estimation of objects based on local invariant features [4, 6, 7, 10], where Lowe's SIFT features [7] are considered state of the art. These features seem to deliver a very stable and accurate pose estimate, but the representation of the local feature is iconic. By using a model based approach to represent these local features, it is possible to have a more compact representation, and it is also possible to extract information about the local area which could be useful in a grouping process.

The approach to pose estimation proposed in this paper uses the scene tensor in 2D [8, 9] as a basis for a set of invariant features. The scene tensor is a representation of one or several line segments, where each segment is represented by its orientation, center of gravity and covariance relative to a local coordinate system. A tensor doublet, inspired by the work of Granlund and Moe [4] and based on the information from the scene tensor is then used as the invariant representation of the local feature. The tensor doublet only consists of four parameters which all are invariant to translation, and variations in orientation and



scale. In comparison with the SIFT feature’s 128 invariant parameters where more than 50 percent is non zero, the tensor doublet is a very low dimensional feature vector. If the database contains a large number of feature vectors, the lower dimensionality of the feature vector will definitely speed up the rest of the pose estimation procedure.

In this paper we will show that it is possible to use low dimensional feature vectors for pose estimation of object and still get comparable results to other approaches using high dimensional feature vectors.

## 2 Introduction to the Scene Tensor

Next section presents a doublet descriptor which is invariant to certain transformations. The descriptor is based on the assumption that we can find corner-like points in the image and also describe the parameters of the corresponding corners; the opening angles, their relative orientation and position. Consequently, we need a detector of corner points and a descriptor of the parameters of the corner. In the literature, the so-called Harris corner detector [5] is a common tool for finding interest points for various purposes. This feature, however, does not give a reliable indication that the corresponding point really is a corner. It also detects isolated points, crossings of several lines, high frequency textures and noise, or in principle anything which cannot be characterized as locally constant or similar to a single line.

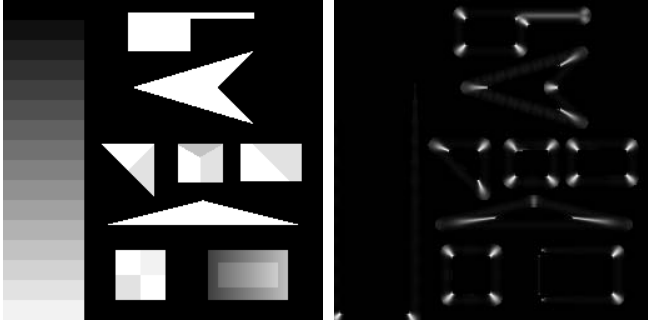
For the purpose of finding and describing corner points we here use a tensor based descriptor presented in [8, 9]. This approach combines the usual orientation tensor [2] with projective geometry, similar to what was done in [11], but also employing higher order tensors (4th order). The resulting representation has some very useful properties presented below.

The fourth order tensor can be rearranged as a second order tensor  $\mathbf{S}_{22}$  which in the 2D case is  $6 \times 6$ . In [8] it is shown that  $\mathbf{S}_{22}$  can be estimated from image data using only weighted polynomial filters which in addition can be separable. Assuming that  $\mathbf{S}_{22}$  has been estimated from a local region which contains  $N$  line segments it can be written

$$\mathbf{S}_{22} = \sum_{k=1}^N \mathbf{S}_{20,k} \mathbf{S}_{02,k}^T \quad (1)$$

where  $\mathbf{S}_{20,k}$  is 6-dimensional vector which contains information about the *local* center of gravity and extension of segment  $k$  and  $\mathbf{S}_{02,k}$  is a 6-dimensional vector which contains information about the position and orientation of the corresponding line. In brief, each pair  $\mathbf{S}_{20,k}$ ,  $\mathbf{S}_{02,k}$  describes a local line segment in terms of its position relative to the local region, its orientation and extension. Consequently,  $\mathbf{S}_{22}$  is the superposition of all this information for all  $N$  line segments in the local region from where it has been estimated.

In [8] it is shown that for the case  $N \leq 3$  the rank of  $\mathbf{S}_{22}$  is the same as  $N$ , i.e., local regions which contains two line segments are characterized by  $\mathbf{S}_{22}$  having



**Fig. 1.** Left: test image. Right: A measurement of certainty that the corresponding  $\mathbf{S}_{22}$  is of rank two.

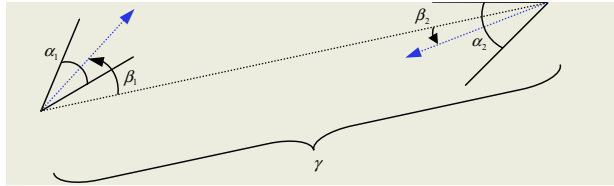
rank two. The detection of rank two points can be implemented in different ways, but it is typically based on analyzing the singular values of  $\mathbf{S}_{22}$ , see [8]. In the 2D case, and for these rank two points, it is also possible to analyze  $\mathbf{S}_{22}$  further to extract the position, extension and orientation of both line segments. This is done by analyzing the full SVD of  $\mathbf{S}_{22}$ . This information allows us to characterize each rank two region, e.g., as a corner, a crossing or a junction. Furthermore, for each of these cases, the position where the two lines meet and the orientations of each line can be estimated with an accuracy which make the representation useful for practical applications, e.g., the one presented in this paper. For a more detailed presentation of the fourth order tensor representation, see [8].

Figure 1 shows a synthetic test image and the response of a rank two measure. By considering the local peaks of the response image, and further analysis of the corresponding  $\mathbf{S}_{22}$  at these points, it is possible to determine if the points are corners and what the parameters of the corners are.

### 3 Compact and Invariant Representation of Local Image Data

One very useful feature for a representation is its degree of invariance. If a representation is invariant with respect to translation, rotation and scale, the amount of training data required decreases and a learning procedure converges faster. The scene tensor described in section 2 is not invariant with respect to orientation and scale and we have consequently implemented an invariant representation based on both the scene tensor and the work by Granlund and Moe [4]. The idea is to calculate invariant parameters based on a geometry including two corners. This representation is called a doublet or more precisely a tensor doublet, because the corners are detected and represented by using the scene tensor.

From the corner feature detection process described in Section 2 we get a list of tensors where each tensor is a description of a local region containing

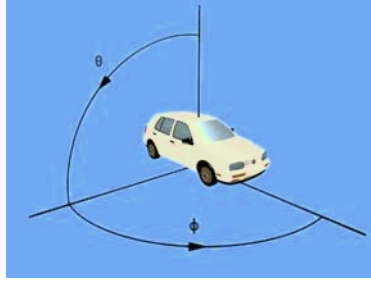


**Fig. 2.** Illustration of the tensor doublet.

two line segments. Each segment is defined in terms of its position, extension and orientation. By extracting the line parameters from two of these tensors the tensor doublet illustrated in figure 2 can be computed. The four feature parameters  $\alpha_1$ ,  $\alpha_2$ ,  $\beta_1$ ,  $\beta_2$  and  $\gamma$  can be calculated from the line parameters, where  $\alpha$  is the angle between the line segments in each second level feature and  $\beta$  is the orientation of each feature relative to the line connecting both features. These four parameters are invariant to both rotation, translation and scaling of the image. The position of each feature is defined by the intersection of the line segments and  $\gamma$  is the distance between the features. It is more robust to use the intersection as the feature's position rather than using the result from the detection process, since that is dependent on contrast, lighting and even the angle between the line segments. It is not hard to realize that the parameters representing the tensor doublet are invariant with respect to translation and rotation. The  $\gamma$  parameter is however not invariant to scale, but is useful in the grouping process. The process of connecting second level features is not an easy task and it is necessary to include some kind of perceptual grouping process in this step. The method employed here is on the lower end of perceptual grouping, where the rule for connecting two features is simply based on the distance  $\gamma$  between the features. If the distance for a feature pair is between certain lower and an upper bounds, then the features are joined to build a doublet. The maximal distance should be set to a value that minimizes the probability of a connection between features from the object and the background. A typical value is half the object size. The minimal distance should prevent connecting two tensors estimated from the same feature and the value should be based on the parameters used in the detection process.

## 4 Mapping from the Representation to Object State Parameters

In this approach to pose estimation we have used a matching and clustering procedure to perform the mapping from feature vectors to object state parameters, but it is also possible to use an associative network together with these types of features. The object state parameters are the two pose angles  $\phi$  and  $\theta$ , figure 3, the scaling relative to the training view  $s'/s$ , the rotation in the image plane  $\alpha$ , and the translation  $x$ ,  $y$ .



**Fig. 3.** Definition of the two pose angles.

During training, images are taken from different views of the object using a rotation table. Tensor doublets are calculated for each image and stored in a database, called the prototype doublets in figure 4. A label containing the pose angles,  $\phi$  and  $\theta$ , together with the positions for the interest points in the doublet is also stored for each prototype doublet. When a query image, or a test image, is presented to the system, tensor doublets are calculated. These doublets are referred to as query doublets in figure 4. Each of these query duplets is then matched with the prototype doublets and for each match a translation  $\mathbf{t}$ , rotation  $\mathbf{R}$  and scaling  $s$  of the object relative to the training image is calculated according to

$$\mathbf{p}_q = \mathbf{t} + s\mathbf{R}\mathbf{p}_p \quad (2)$$

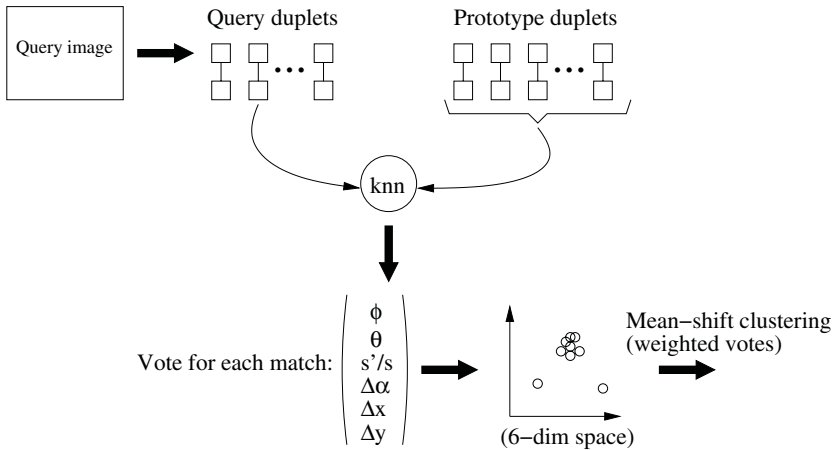
where  $\mathbf{p}_q$  and  $\mathbf{p}_p$  are the positions of one of the points in the doublet in the query image and in the prototype image, respectively. The transformations have 4 degrees of freedom in total, so one doublet should be sufficient to compute the transformations.

All doublets computed from interest points on the objects will vote on the same object state parameters and will therefore cluster in the six dimensional space illustrated in figure 4. This cluster can be found by a mean-shift filtering followed by a mean-shift clustering [1, 3]. A confidence measurement is then calculated for each cluster. This measure is the estimated probability for the cluster mean multiplied with the number of votes in the cluster.

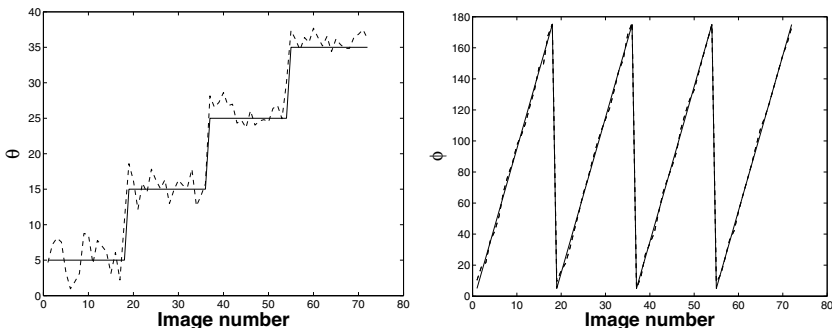
## 5 Results

The system has been trained for one of the sockets in figure 6. Images have been taken from different views of this object where  $\theta$  ranges from zero to  $40^\circ$  and  $\phi$  ranges from zero to  $180^\circ$ . The step between the training images is  $10^\circ$  for both the  $\phi$  and  $\theta$  variable.

The pose estimation system has been evaluated with the worst case images, meaning the images between the training images. The result is illustrated in figure 5. The MAE (mean absolute error) is  $1.6^\circ$  for the  $\theta$  variable and  $1.8^\circ$  for  $\phi$ .

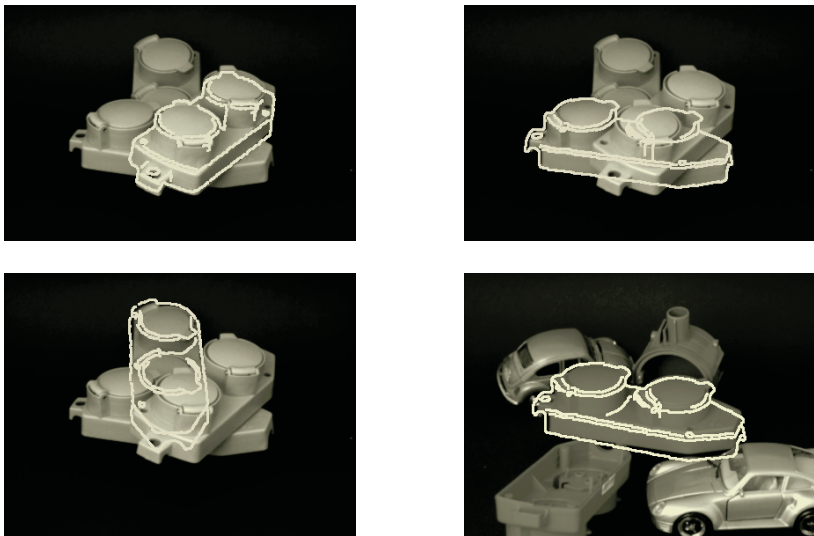


**Fig. 4.** Overview of the query mode. The resulting output is an estimated pose, position, rotation, and scale of the object. KNN refers to the  $k$  nearest neighbor method.



**Fig. 5.** Pose estimation test on the socket. The dotted line is the estimated pose and the solid line represents the ground truth.

The bin picking problem discussed in the introduction often implies in practice that the objects are stacked in a pile. The pose estimation system is evaluated for such a situation with good performance, figure 6. Each one of the three objects can be found with good accuracy. The upper leftmost image illustrates the pose derived from the cluster with the highest confidence, the upper rightmost is the cluster with the second highest confidence and the lower leftmost is the cluster with the third highest confidence. The white mesh illustrating the object pose is the norm of the gradient of the closest training view, which is scaled and translated according to the pose estimate. The system also works with other objects in the background which is illustrated in the lower rightmost image in figure 6. Figure 7 illustrates the performance when the object has a different scale relative to the training images.



**Fig. 6.** Pose estimation of sockets in a pile and a socket with background. The three first images is actually the same image, where the first image illustrates the pose estimate with the highest confidence, the second image illustrates the pose estimate with the second highest pose estimate and so on. Demonstrably the algorithm can detect several objects from one image.



**Fig. 7.** Pose estimation with background and different scales.

## 6 Conclusion

In this paper we have introduced the tensor doublet, which is a compact and invariant representation useful for pose estimation tasks. The main difference between this representation and others is the low dimension of the feature vector, which will definitely speed up the following steps in the pose estimation algorithm. It is shown by a number of test images that the pose estimation works well for objects stacked in a pile, an object with a cluttered background and objects with different scales. Pose estimation of objects stacked in a pile is especially interesting in industrial automation, for example the bin picking problem.

## References

1. Yizong Cheng. Mean shift, mode seeking, and clustering. 17(8):790–799, August 1995.
2. Gunnar Farneback. *Polynomial Expansion for Orientation and Motion Estimation*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 2002. Dissertation No 790, ISBN 91-7373-475-6.
3. Keinosuke Fukunaga and Larry D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.
4. Gösta H. Granlund and Anders Moe. Unrestricted recognition of 3-D objects for robotics using multi-level triplet invariants. *Artificial Intelligence Magazine*, 25(2):51–67, 2004.
5. C. G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, September 1988.
6. Björn Johansson and Anders Moe. Patch-duplets for object recognition and pose estimation. Technical Report LiTH-ISY-R-2553, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, November 2003.
7. David G. Lowe. Local feature view clustering for 3D object recognition. In *Proc. CVPR'01*, 2001.
8. Klas Nordberg. A fourth order tensor for representation of orientation and position of oriented segments. Technical Report LiTH-ISY-R-2587, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, May 2004.
9. Klas Nordberg and Robert Söderberg. Detection and estimation of features for estimation of position. In Ewert Bengtsson and Mats Eriksson, editors, *Proceedings SSBA'04 Symposium on Image Analysis*, pages 74–77, Uppsala, March 2004. SSBA.
10. C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.
11. S. M. Seitz and P. Anandan. Implicit representation and scene reconstruction from probability density functions. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 28–34, 1999.

# Gabor Parameter Selection for Local Feature Detection\*

Plinio Moreno, Alexandre Bernardino, and José Santos-Victor

Instituto Superior Técnico & Instituto de Sistemas e Robótica  
1049-001 Lisboa - Portugal  
{plinio,alex,jasv}@isr.ist.utl.pt

**Abstract.** Some recent works have addressed the object recognition problem by representing objects as the composition of independent image parts, where each part is modeled with “low-level” features. One of the problems to address is the choice of the low-level features to appropriately describe the individual image parts. Several feature types have been proposed, like edges, corners, ridges, Gaussian derivatives, Gabor features, etc. Often features are selected independently of the object to represent and have fixed parameters. In this work we use Gabor features and describe a method to select feature parameters suited to the particular object considered. We propose a method based on the Information Diagram concept, where “good” parameters are the ones that optimize the filter’s response in the filter parameter space. We propose and compare some concrete methodologies to choose the Gabor feature parameters, and illustrate the performance of the method in the detection of facial parts like eyes, noses and mouths. We show also the rotation invariance and robustness to small scale changes of the proposed Gabor feature.

## 1 Introduction

The object recognition problem has been tackled recently with several successful results [1–4]. All of these works exploit the idea of selecting various points in the object and building up a local neighborhood representation for each one of the selected points. Two related problems are involved in this process: (i) which points in the object should be used and (ii) how to represent the information contained in their neighborhood. In the present work, we address the latter problem, assuming that interest points are obtained by some methodology [1–3]. In the experiments we present later, interest points are selected manually.

Regarding the problem of local neighborhood representation, there are several types of features being proposed in the literature: gradient magnitude and orientation maps [1], Gaussian derivatives [2, 3], rectangular features [5] and

---

\* Research partly funded by European project IST 2001 37540(CAVIAR), the FCT Programa Operacional Sociedade de Informação(POSI) in the frame of QCA III, and Portuguese Foundation for Science and Technology PhD Grant FCT SFRH\BD\10573\2002.



Gabor features[6], amongst others. However, the parameters when using Gabor features are often fixed [7, 8] or chosen manually [6]. In this work we also use Gabor features to represent a local image neighborhood but select their parameters according to the particular image pattern to detect.

The adaptation of feature parameters to particular object parts was first exploited in [9]. They propose to select Gabor function parameters in a semi-automatic fashion, using the “Information Diagram” concept. The Information Diagram is the representation of Gabor feature magnitude, at a certain image point, as a function of the Gabor filter orientation and frequency parameters. The scale and wavelength (inverse of frequency) have a fixed ratio. In our work, we extend the “Information Diagram” concept to consider scale and wavelength as independent parameters, thus resulting in a 3-dimensional function. We show different methodologies to select “good” feature parameters from this Extended Information Diagram.

In order to evaluate different methodologies for parameter selection, we have set-up a facial feature learning and detection experiment. The evaluation of results will be based on the detection rates achieved. Since the focus of the work is on the selection of feature parameters, we will employ very straightforward techniques for the learning and detection steps. In the learning step we compute the object model, consisting in the average and covariance matrix of vectors containing the response of selected Gabor features in a large training set. In the detection step, we compute the distance (Euclidean and Mahalanobis) between novel image points and the acquired models. We have performed experiments in the identification of facial points like eyes, mouths and noses, and obtain high success rates with the proposed features. Then we evaluate the robustness of the method to pattern variations in scale and orientation.

## 2 Gabor Functions as Local Image Descriptors

The motivation to use Gabor functions is mostly biological, since Gabor-like receptive fields have been found in the visual cortex of primates [10]. Gabor functions act as low-level oriented edge and texture discriminators and are sensitive to different frequencies and scale information. These facts raised considerable interest and motivated researchers to extensively exploit the properties of Gabor functions.

### 2.1 The Gabor Function

Mathematically, a 2D Gabor function,  $g$ , is the product of a 2D Gaussian and a complex exponential function. The general expression is given by:

$$g_{\theta,\lambda,\sigma_1,\sigma_2}(x,y) = \exp\left\{-1/2(x\ y)M(x\ y)^T\right\} \exp\left\{\frac{j\pi}{\lambda}(x\cos\theta + y\sin\theta)\right\}$$

where  $M = \text{diag}(\sigma_1^{-2}, \sigma_2^{-2})$ . Some examples of Gabor functions are shown in Fig.1. The parameter  $\theta$  represents the orientation,  $\lambda$  is the wavelength, and

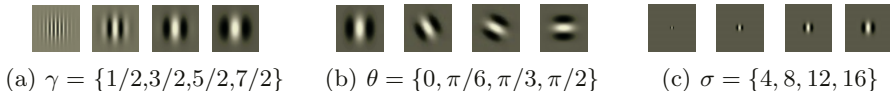
$\sigma_1$  and  $\sigma_2$  represent scale at orthogonal directions. When the Gaussian part is symmetric, we obtain the isotropic Gabor function:

$$g_{\theta,\lambda,\sigma}(x,y) = \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\} \exp\left\{\frac{j\pi}{\lambda}(x\cos\theta+y\sin\theta)\right\} \quad (1)$$

However, with this parameterization the Gabor function does not scale uniformly, when  $\sigma$  changes. It is preferable to use a parameter  $\gamma = \lambda/\sigma$  instead of  $\lambda$  so that a change in  $\sigma$  corresponds to a true scale change in the Gabor function. Also, it is convenient to apply a 90 degrees counterclockwise rotation to Eq. (1), such that  $\theta$  expresses the orthogonal direction to the Gabor function edges. Therefore, in the remainder of the paper we will use the following definition for the Gabor functions:

$$g_{\theta,\gamma,\sigma}(x,y) = \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\} \exp\left\{\frac{j\pi}{\gamma\sigma}(x\sin\theta-y\cos\theta)\right\}$$

By selectively changing each of the Gabor function parameters, we can “tune” the filter to particular patterns arising in the images. In Fig. 1 we illustrate the variation of parameters( $\gamma$ ,  $\theta$ ,  $\sigma$ ) in the shape of the Gabor function.



**Fig. 1.** Examples of Gabor functions. Each sub-figure shows the real part of Gabor function for different values of  $\gamma$ ,  $\theta$ , and  $\sigma$

## 2.2 Gabor Response

By convolving a Gabor function with image patterns  $I(x,y)$ , we can evaluate their similarity. We define the Gabor response at point  $(x_0, y_0)$  as:

$$G_{\theta,\gamma,\sigma}(x_0, y_0) = (I * g_{\theta,\gamma,\sigma})(x_0, y_0) = \int I(x,y)g_{\theta,\gamma,\sigma}(x_0-x, y_0-y)dx dy \quad (2)$$

where  $*$  represents convolution. The Gabor response obtained from Eq. (2) can emphasize basically three types of characteristics in the image: edge-oriented characteristics, texture-oriented characteristics and a combination of both. In order to emphasize different types of image characteristics, we must vary the parameters  $\sigma$ ,  $\theta$  and  $\gamma$  of the Gabor function.

The variation of  $\theta$  changes the sensitivity to edge and texture orientations. The variation of  $\sigma$  will change the “scale” at which we are viewing the world, and the variation of  $\gamma$  the sensitivity to high/low frequencies. We would like to find the most adequate combinations of  $\sigma$ ,  $\theta$  and  $\gamma$  to represent particular parts of objects for recognition/detection tasks.

### 3 Object Part Model

As mentioned in the introduction, we consider objects composed of parts, like eyes, noses and mouths in human faces. Each part is modeled as random vector containing (a) the absolute value of Gabor responses, and (b) the real and imaginary parts of Gabor responses with different parameters. In the case of (a), the feature vector is:

$$v_{(x,y)} = \left( v_{(x,y)}^1, \dots, v_{(x,y)}^i, \dots, v_{(x,y)}^m \right)^T; \quad v_{(x,y)}^i = |G_{\theta_i, \gamma_i, \sigma_i}(x, y)| \quad (3)$$

and  $(x, y)$  represents the coordinate of the object part center. In the case of (b), the feature vector is:

$$v = \left( v^1, \dots, v^i, \dots, v^{2m} \right)^T; \quad v^{2i} = \text{Re}(G_{\theta_i, \gamma_i, \sigma_i}); \quad v^{2i-1} = \text{Im}(G_{\theta_i, \gamma_i, \sigma_i}) \quad (4)$$

The rationale is to model image parts by analyzing their contents in terms of edges and textures of different scales, orientations and frequencies. We assume that the random feature vector follows a normal distribution with average  $\bar{v}$  and covariance matrix  $\Sigma$ ,  $v_{(x,y)} \sim \mathcal{N}(\bar{v}_{(x,y)}, \Sigma_{(x,y)})$ .

For the detection of parts, we will compute the distance between the obtained model and the novel patterns. We consider both the Euclidean and Mahalanobis distances. The decision of whether a part feature is present or not in a certain image pixel will depend on the computed distance values.

## 4 Parameter Selection

In this section we focus on selecting the parameters (orientation, scale and frequency) for each of the Gabor functions used in the feature model. We assume a limited (constant) number of Gabor filters to constrain the computational cost of the methods. A straightforward approach to define the parameters would be to sample the parameter space uniformly. However, this strategy does not exploit the particular characteristics of the object part under test. Instead, we could analyse the Gabor response function in the full parameter space  $(\sigma, \gamma, \theta)$  and select those parameters that best describe the particular object characteristics. However, this strategy could bias the parameter distribution to a narrow range and reduce the capability to discriminate the modeled object from others. To enforce some variability in the parameter space and still be able to adapt the representation to the particular object under test, we will sample uniformly one of the parameters and perform a 2D search in the remaining dimensions. This strategy extends the concept of **Information Diagram**[9].

### 4.1 Information Diagram

The ‘‘Information Diagram’’ (ID) concept proposed in [9] selects the Gabor filter parameters semi-automatically. The ID represents the magnitude of the Gabor

response at a certain interest point  $(x, y)$ , as a function of  $\theta$  and  $\sigma$ , keeping the  $\gamma$  parameter constant. The ID function is defined as:

$$\text{ID}_{x,y}(\theta, \sigma) = |G_{\theta, \gamma=1, \sigma}(x, y)|$$

Then, local maxima coordinates of ID are chosen as “good” Gabor function’s parameters because they represent the object’s characteristic orientations, scales and frequencies, thus being considered good descriptors of the local image content.

In this work, we extend the ID concept to consider variability also in the  $\gamma$  parameter. We define the Extended Information Diagram as the 3D function:

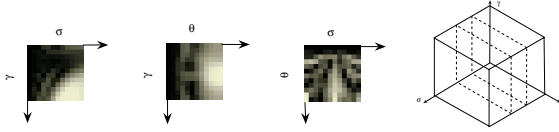
$$\text{EID}_{x,y}(\theta, \sigma, \gamma) = |G_{\theta, \gamma, \sigma}(x, y)|$$

Then we denote  $\theta$ -ID,  $\gamma$ -ID, and  $\sigma$ -ID as slices of the EID function, keeping constant one of the parameters,  $\theta = \theta_0$ ,  $\gamma = \gamma_0$  or  $\sigma = \sigma_0$ :

$$\theta\text{-ID}_{x,y}^{\theta_0}(\sigma, \gamma) = \text{EID}_{x,y}(\theta_0, \gamma, \sigma); \quad \sigma\text{-ID}_{x,y}^{\sigma_0}(\theta, \gamma) = \text{EID}_{x,y}(\theta, \gamma, \sigma_0);$$

$$\gamma\text{-ID}_{x,y}^{\gamma_0}(\theta, \sigma) = \text{EID}_{x,y}(\theta, \gamma_0, \sigma)$$

According to our notation, the work in [9] uses a  $\gamma$ -ID with  $\gamma_0 = 1$ . In Fig. 2 we show some examples of the  $\theta$ -ID,  $\sigma$ -ID and  $\gamma$ -ID.



**Fig. 2.** Examples of  $\theta$ -ID,  $\sigma$ -ID,  $\gamma$ -ID, and  $\sigma$  slices in the parameter space from left to right

## 4.2 Searching Multiple Information Diagrams

Our strategy to find good parameters for the object part’s model features is based on uniformly discretizing one of the parameters (say  $\theta$ ), and search local extrema in the resulting set of  $\theta$ -ID’s. For example, a set of  $\theta$ -IDs for  $\mathcal{T} = \{\theta_1, \dots, \theta_n\}$ , at point  $(x, y)$  is given by:

$$\Theta\text{-ID}_{x,y}^{\mathcal{T}} = \{\theta\text{-ID}_{x,y}^{\theta_1}, \dots, \theta\text{-ID}_{x,y}^{\theta_n}\}$$

The several  $\theta_i \in \mathcal{T}$  are uniformly spaced in the range  $[0, \pi)$ . Then we compute the parameters of the highest local maximum and smallest local minimum:

$$(\hat{\sigma}_i^{\max}, \hat{\gamma}_i^{\max}) = \arg \max_{\sigma, \gamma} \theta\text{-ID}_{x,y}^{\theta_i}; \quad (\hat{\sigma}_i^{\min}, \hat{\gamma}_i^{\min}) = \arg \min_{\sigma, \gamma} \theta\text{-ID}_{x,y}^{\theta_i}$$

Then, the set of chosen Gabor function parameters in Eq.(3) and Eq.(4), are such that  $(\gamma_i, \theta_i, \sigma_i)$  belongs to  $\{(\hat{\sigma}_1^{\min}, \hat{\gamma}_1^{\min}, \theta_1), \dots, (\hat{\sigma}_n^{\min}, \hat{\gamma}_n^{\min}, \theta_n)\}$  and/or  $\{(\hat{\sigma}_1^{\max}, \hat{\gamma}_1^{\max}, \theta_1), \dots, (\hat{\sigma}_n^{\max}, \hat{\gamma}_n^{\max}, \theta_n)\}$ .

## 5 Experimental Results

In this section we present the results of the tests done for the various approaches to object modeling and feature parameter selection. Then we select the most successful approach and perform tests in order to verify the rotation invariance and robustness to scale changes of the selected feature vector.

The experimental tests performed in this work use 90 subjects from the AR face database [11], all without glasses, where 60 of them are used for training (object part modeling) and 30 for testing (object part detection). We represent four different parts: left eye, right eye, nose and mouth.

### 5.1 Selection of the Object Model and the Modified ID

Experiments are set-up for evaluating the discretized parameters ( $\sigma$ ,  $\gamma$  or  $\theta$ ), the number and type of the extrema computed at each ID, the distance metrics (Euclidean and Mahalanobis), and the feature model type (magnitude *vs* real-imaginary parts). A list of the experiments and related configurations is shown in Table 1.

**Table 1.** List of the performed tests. Performance in last two columns(%)

Test ID	type	# local max	# local min	distance	mag	re+im
1	$\theta$	1	1	Mah	68.49	78.33
2	$\theta$	2	0	Mah	85.92	95.83
3	$\gamma$	2	0	Mah	58.19	74.16
4	$\gamma$	1	1	Mah	54.41	75.83
5	$\sigma$	2	0	Mah	58.19	72.50
6	$\sigma$	1	1	Mah	50.21	72.50
7	$\theta$	1	1	Euc	31.93	85
8	$\theta$	2	0	Euc	38.87	87.5
9	$\gamma$	2	0	Euc	17.86	53.33
10	$\gamma$	1	1	Euc	15.55	45
11	$\sigma$	2	0	Euc	24.79	74.17
12	$\sigma$	1	1	Euc	15.97	75.83

In every experiment performed we use  $n = 12$  IDs, and choose either one local maxima and one local minima or two local maxima, so the number of filters is kept constant ( $m = 24$ ). The sets of values for the  $\theta$ ,  $\gamma$  and  $\sigma$ -IDs are, respectively,  $\mathcal{T} = \{0, \pi/12, \dots, 11\pi/12\}$ ,  $\mathcal{G} = \{0.5, 0.8, \dots, 4\}$ , and  $\mathcal{S} = \{4, 7, \dots, 39\}$ .

All IDs are calculated from the mean images  $\bar{I}_{\text{part}}$  in the training set, centered at each object part (left eye, right eye, nose, mouth). To evaluate the performance of each experiment we count the number of hits (successful detections) in the test set. Given an object part model, a distance function and an image point, a hit exists if the global minima of the distance is located inside a circle of radius  $r$  around the center of the object part.

Considering only the tests using real and imaginary parts of the Gabor response, we can see, in Fig. 3 that mean performance is better when using  $\theta$ -IDs, Mahalanobis distance, and 2 local maxima. In this case, success rates are as high as 95%.

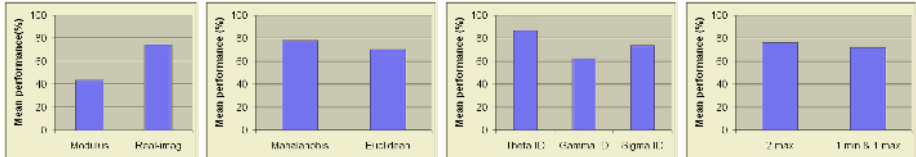


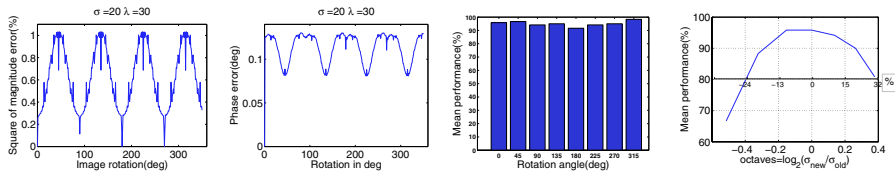
Fig. 3. Mean detection rate of marginalized tests of table 1

## 5.2 Rotation Invariance

We test the rotation invariance of the Gabor filters on a synthetic image, and evaluate, in the face data set, the effects of Gabor response variations to rotated patterns on the correct detection rate. Due to discretization effects and imperfect filter symmetry, Gabor response presents small variations with the amount of rotation. In Fig.4 we show the effect of image  $\alpha$ -rotation in the response of a Gabor  $\alpha$ -rotated to a synthetic image at the image’s center point. We can observe that there are some errors in the magnitude and phase that, not being dramatic, can change the performance of the detection algorithm. The variation in the error change the success rate in the object part model when using rotated images. If we shift the angles in every component of the feature vector in Eq.(4), the rotated model is:  $v_{(x,y)} = (v^1, \dots, v^i, \dots, v^{2m})^T$ ;  $v_{(x,y)}^{2i} = \text{Re}(G_{\theta_i+\alpha, \lambda_i, \sigma_i}(x, y))$ ;  $v_{(x,y)}^{2i-1} = \text{Im}(G_{\theta_i+\alpha, \lambda_i, \sigma_i}(x, y))$ . In Fig. 4 we observe the variation of the success rate when rotating the image parts and the model. In our tests, for simplicity, we rotate the image regions every  $\pi/4$ , because it does not involve a recomputation of the filters response. It is important to remark that we use the object model learned when  $\alpha = 0$ , computed in the previous section for test 2 in Table 1. We observe a very good behaviour of the learned model in the rotated images, with a performance above 91%.

## 5.3 Scale Robustness

To check the robustness to scale variations, we compute the success rate in rescaled images maintaining the object model learned in the original images ( $\theta$ -IDs, Mahalanobis distance, and two local maxima). In Fig. 4 we observe that the performance is above 90% for image rescaling upto  $\pm 20\%$ , corresponding to a range of about 0.6 octaves. To cope with larger scale variations, one should cover the scale dimension with additional object part models. If we sample the scale space every 0.6 octaves we should be able to keep performance above 90%, provided that an adequate scale selection method is available.



**Fig. 4.** Gabor filter rotation invariance tests(magnitude error, phase error, and success rate variation in rotated images) and scale robustness test, from left to right

## 6 Conclusions and Future Work

In this work we present an automatic feature selection method that can be applied to different image regions successfully. The representation is based on Gabor features and our methodology selects automatically a set of parameters that are good descriptors for a particular image pattern, representing a part of an object. The technique is based on the Information Diagram concept [9], that is extended, in this work, to consider optimization along all dimensions of the Gabor function parameters. We illustrate the richness of the descriptor and parameter selection methods in a facial feature detection task.

The face detection tests allowed us to evaluate certain design criteria:

- a representation using the full Gabor response (real and imaginary parts) is more powerful than using the magnitude alone;
- using  $\theta$ -ID’s provided significantly better performance;
- the Mahalanobis distance outcomes the Euclidean distance in the detection success;

We also show some tests illustrating the rotation and scale robustness characteristics of the method. The detection method is based on simple distance metrics to stress the feature capability in representing image patterns, independently of sophisticated learning algorithms. Even though the learning algorithm is very simple, results are promising and should further improve with more powerful techniques.

## References

1. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (1999) 1150–1157
2. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19** (1997) 530–534
3. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: *European Conference on Computer Vision*, Springer (2002) 128–142
4. Weber, M., Welling, M., Perona, P.: Towards automatic discovery of object categories. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2000)
5. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2001)

6. Smeraldi, F., Bigun, J.: Retinal vision applied to facial features detection and face authentication. *Pattern Recognition Letters* **23** (2002)
7. Jain, A., Ratha, N., Lakshmanan, S.: Object detection using gabor filters. *Pattern Recognition* **30** (1997) 295–309
8. Wu, H., Yoshida, Y., Shioyama, T.: Optimal gabor filters for high speed face identification. In: 16th International Conference on Pattern Recognition. Volume 1. (2002) 11–15
9. Kamarainen, J.K., Kyrki, V., Kälviäinen, H.: Fundamental frequency gabor filters for object recognition. In: Proc. of the 16th International Conference on Pattern Recognition. (2002)
10. Daniel, P., Whitteridge, D.: The representation of the visual field on the cerebral cortex in monkeys. *Journal of Physiology* **159** (1961) 203–221
11. Martinez, A., Benavente, R.: The ar face database. Technical report, CVC (1998)



# Real-Time Tracking Using Multiple Target Models

Manuel J. Lucena<sup>1</sup>, José M. Fuertes<sup>1</sup>, and Nicolás Pérez de la Blanca<sup>2</sup>

<sup>1</sup> Departamento de Informatica, Escuela Politecnica Superior, Universidad de Jaen  
Campus de las Lagunillas, 23071 Jaen, Spain  
{mlucena, jmf}@ujaen.es

<sup>2</sup> Departamento de Ciencias de la Computacion e Inteligencia Artificial  
ETSII. Universidad de Granada  
C/ Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain  
nicolas@ugr.es

**Abstract.** Using Comaniciu et al.'s approach as a basis, [9], this paper presents a real-time tracking technique in which a multiple target model is used. The use of a multiple model shall enable us to provide the tracking scheme with a greater robustness for tracking tasks on sequences in which there are changes in the lighting of the tracked object. In order to do so, a selection function is defined for the model to be used in the search process of the object in each frame.

## 1 Introduction

Tracking objects through the frames of an image sequence is a critical task in online and offline image-based applications such as surveillance, visual serving, gestural human-machine interfaces, video editing and compression, augmented reality and visual effects, motion capture, driver assistance, medical and meteorological imaging, etc.

Bayesian framework methods have played an important role in tracking [1][2][3]. The inclusion of a prior offline learning phase enables objects with more complicated shapes to be tracked [4][5][6]. Exemplar-based methods generate object representations from examples and then use distance measures to perform template matching.

If the objects to be tracked are non-rigid, it is advisable to represent them with probability distributions. A straightforward way to derive a distribution model is by using histogram analysis [7][8][9]. The techniques introduced independently by Bradski and Comaniciu et al. are based on the following principle: the current frame is searched for a region, a fixed-shape variable-size window, whose color content best matches a reference color model. The search is deterministic. Starting from the final location in the previous frame, it proceeds iteratively at each frame so as to minimize a distance measure to the reference color histogram. Objects are modeled using color distributions and the similarity is then measured between the target and candidate distributions using a Bhattacharyya coefficient.

A key component of a successful tracking system is the ability to search efficiently for the target, as real-time tracking is one of the main goals of our research.

Comaniciu et al. [9] propose a tracking algorithm in which a scheme for object representation and tracking is established from the definition of a single target model. The reference target model is represented by its pdf  $q$  in the feature space. The reference model can be chosen to be the color pdf of the target. In the subsequent frame, a *target candidate* is defined at location  $\mathbf{y}$ , and is characterized by the pdf  $p(\mathbf{y})$ . Both pdfs are estimated from the data. In order to satisfy the low-computational cost imposed by real-time processing discrete densities,  $m$ -bin histograms should be used.

In certain cases, when the target moves in variable lighting conditions, shadows appear which significantly alter the color distributions in the image sequence (Figure 1). A single pdf will therefore be insufficient for modeling and tracking the object reliably. Our approach is based on the use of multiple pdfs in a single target model, when lighting conditions change between frames.

This paper is organized into four sections: Section 2 presents a short review of the multiple model tracking technique; Section 3 presents some experimental results; and finally, Section 4 concludes the paper.



Fig. 1. Three frames of a sequence where the target presents different illumination conditions.

## 2 Tracking

### 2.1 Target Representation

In this section, we shall briefly present the main elements defined by Comaniciu et al. [9] in their tracking scheme. The pdfs defined for the target model and the target candidate will be given by  $m$ -bin histograms.

$$\begin{aligned} \text{target model:} \quad \hat{\mathbf{q}} &= \{\hat{q}_u\}_{u=1\dots m} & \sum_{u=1}^m \hat{q}_u &= 1 \\ \text{target candidate:} \quad \hat{\mathbf{p}}(\mathbf{y}) &= \{\hat{p}_u(\mathbf{y})\}_{u=1\dots m} & \sum_{u=1}^m \hat{p}_u &= 1 \end{aligned}$$

A target is represented by an ellipsoidal region in the image. All targets are first normalized to a unit circle.

The function  $b : R^2 \rightarrow \{1 \dots m\}$  associates to the pixel at location  $\mathbf{x}_i^*$  the index  $b(\mathbf{x}_i^*)$  of its bin in the quantized feature space. The probability of the feature  $u = 1 \dots m$  in the target model is then computed as:

$$\hat{q}_u = C \sum_{i=1}^n k(\|\mathbf{x}_i^*\|^2) \delta[b(\mathbf{x}_i^*) - u] \quad (1)$$

where  $\delta$  is the Kronecker delta function and  $C$  is a normalization constant.

Let  $\{\mathbf{x}_i\}_{i=1 \dots n_h}$  be the *normalized* pixel locations of the target candidate, centered at  $\mathbf{y}$  in the current frame. Using the same kernel profile  $k(x)$ , but with bandwidth  $h$ , the probability of the feature  $u = 1 \dots m$  in the target candidate is given by:

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right) \delta[b(\mathbf{x}_i^*) - u] \quad (2)$$

where  $C_h$  is a normalization constant.

## 2.2 Minimization Algorithm

The similarity function defines a distance between the target model and the candidates. The distance between two discrete distributions is defined as:

$$d(\mathbf{y}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]} \quad (3)$$

where the similarity function will be denoted by:

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}) \hat{q}_u} \quad (4)$$

which is the sample estimate of the Bhattacharyya coefficient between  $\mathbf{p}$  and  $\mathbf{q}$  [10].

In order to find the location corresponding to the target in the current frame, the distance (3) should be minimized as a function of  $\mathbf{y}$ . This is equivalent to maximizing the Bhattacharyya coefficient  $\hat{\rho}(\mathbf{y})$ . For this, Comaniciu et al. [9] use the mean-shift algorithm with a monotone kernel.

## 2.3 Model Selection

In order to prevent losses of the target due to lighting changes, we propose a multiple model  $\mathbf{M}$ , comprising a set of  $n$  pdfs, corresponding to several different histograms of the object under typical lighting conditions:

$$\mathbf{M} = \{\hat{\mathbf{q}}_0, \hat{\mathbf{q}}_1, \dots, \hat{\mathbf{q}}_{n-1}\} \quad (5)$$

Running the target localization algorithm for each  $\hat{\mathbf{q}}_i$ , we obtain a set  $\mathbf{B}$  of Bhattacharyya coefficients,

$$\mathbf{B} = \{b_0, b_1, \dots, b_{n-1}\}$$

and a set  $\mathbf{Y}$  of image positions

$$\mathbf{Y} = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{n-1}\}$$

representing the best target location for each model and the corresponding similarity levels.

We then need to select the pdf in  $\mathbf{M}$  which best fits the observed frame. Selecting the one with the largest Bhattacharyya coefficient may increase the risk of distractions with image regions having similar histograms to the ones present in our model. In order to avoid this, we shall also take into account the position of the maximum given for each  $\hat{\mathbf{q}}_i$ , and define a probability distribution based on the difference between the position  $\mathbf{y}_i$  estimated by the tracker, and the predicted position  $\bar{\mathbf{y}}$  of the target. A value of  $\bar{\mathbf{y}}$  for each frame can be obtained by using a dynamical model of the object to be tracked.

Assuming statistical independence between  $\mathbf{B}$  and  $\mathbf{Y}$ , we can define the probability of each  $\hat{\mathbf{q}}_i$ , given  $\mathbf{B}$  as:

$$p(\hat{\mathbf{q}}_i/\mathbf{B}) = \frac{b_i \cdot p(\hat{\mathbf{q}}_i)}{\sum_j (b_j \cdot p(\hat{\mathbf{q}}_j))} \quad (6)$$

with  $p(\hat{\mathbf{q}}_i)$  being the *a priori* probability distribution for each pdf in  $\mathbf{M}$ . Additionally, the probability of each  $\hat{\mathbf{q}}_i$ , given  $\mathbf{Y}$ , is given by:

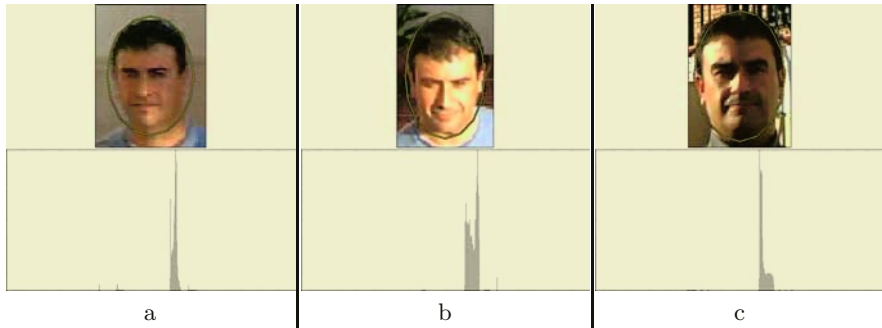
$$p(\hat{\mathbf{q}}_i/\mathbf{Y}) = \frac{p(\bar{\mathbf{y}} - \mathbf{y}_i) \cdot p(\hat{\mathbf{q}}_i)}{\sum_j (p(\bar{\mathbf{y}} - \mathbf{y}_j) \cdot p(\hat{\mathbf{q}}_j))} \quad (7)$$

In our case, we suppose that the  $\bar{\mathbf{y}} - \mathbf{y}_i$  values follow a zero-mean Gaussian distribution, i.e.  $p(\bar{\mathbf{y}} - \mathbf{y}_i) \sim N(0, \sigma)$ . Expressions (6) and (7) lead us to the probability distribution used to select the best pdf for each frame:

$$\begin{aligned} p(\hat{\mathbf{q}}_i/\mathbf{B}, \mathbf{Y}) &= \frac{p(\hat{\mathbf{q}}_i/\mathbf{B}) \cdot p(\hat{\mathbf{q}}_i/\mathbf{Y})}{p(\hat{\mathbf{q}}_i)} \\ &= \frac{b_i \cdot p(\bar{\mathbf{y}} - \mathbf{y}_i) \cdot p(\hat{\mathbf{q}}_i)}{\sum_j (b_j \cdot p(\hat{\mathbf{q}}_j)) \cdot \sum_j (p(\bar{\mathbf{y}} - \mathbf{y}_j) \cdot p(\hat{\mathbf{q}}_j))} \end{aligned} \quad (8)$$

### 3 Results

We have tested the efficiency of our method based on multiple target models by comparing it with a mean-shift tracker using single models [9] with different sequences and lighting conditions. We have used a three-component multiple model containing the simple models shown in Figure 2, and compared the obtained results. All of the experiments have been carried out on a desktop PC (Pentium IV at 2 GHz), at real-time speed (over 40 fps).



**Fig. 2.** Regions used to calculate the RGB models, with their correspondent histograms. The corresponding images belong to different sequences that the ones used for the experiments.



**Fig. 3.** Test Sequence 1, tracking with RGB histograms. a), b) and c): simple models shown in Figure 2; d): multiple model (on the upper-left corner of the images, the best model for that frame is shown).

In this paper, we show the application of the mean-shift tracker both for a simple and a multiple model, on three different sequences. In order to compute the  $m$ -bins histograms required for the tracking algorithm, two color spaces have been used: RGB quantized into  $8 \times 8 \times 8$  bins, and YUV quantized into  $16 \times 4 \times 4$ , obtaining histograms with the same number of bins, but which are more sensitive to intensity in the second case.



**Fig. 4.** Test Sequence 2, tracking with RGB histograms. a), b) and c): simple models shown in Figure 2; d): multiple model (on the upper-left corner of the images, the best model for that frame is shown).

The sequences represent a person who is moving in different directions, moving closer and farther away (scale changes) and varying the speed of the movement. As a result of the presence of shadows in two of the sequences, there are changes in the lighting of the target on entering or leaving these (see Figure 1).

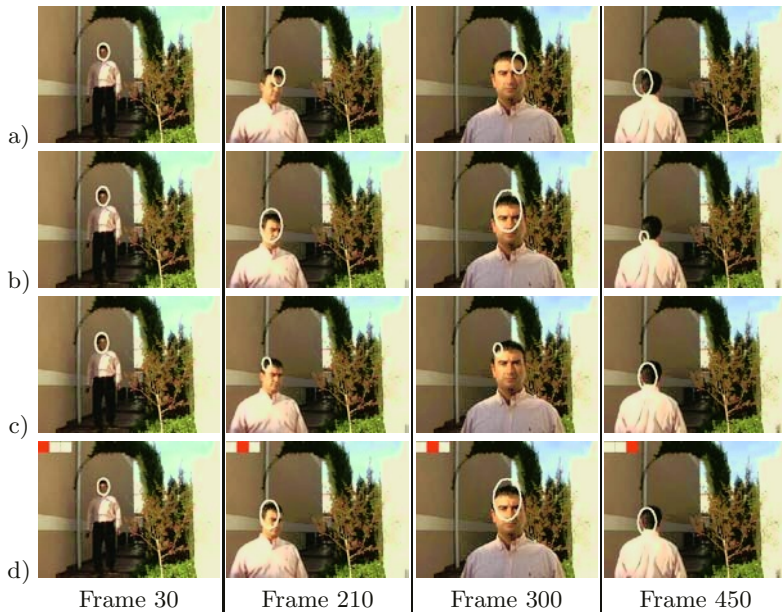
The tracking carried out in the sequences is defined on an ellipsoidal region covering the face. The model was obtained using different images, representing the target in different lighting conditions. Consequently, the target models used for the experiments do not belong to the test sequences. This is an advantage since the multiple model may be initialized offline by employing the set of images that best match the illumination conditions of the sequence.

In Figure 2, the images used to calculate the models are shown, together with their corresponding histograms, weighted with an Epanechnikov kernel of the type used in [9].

In order to predict the position  $\bar{\mathbf{y}}^{t+1}$  of the target in the next frame, a very simple dynamics has been used:

$$\begin{aligned} \mathbf{d}^{t+1} &= \lambda \cdot (\mathbf{y}^t - \mathbf{y}^{t-1}) + (1 - \lambda) \cdot \mathbf{d}^t \\ \bar{\mathbf{y}}^{t+1} &= \mathbf{y}^t + \mathbf{d}^{t+1} \end{aligned} \quad (9)$$

where  $\mathbf{y}^t$  represents the position of the target obtained by the tracking algorithm at time  $t$ , and  $\mathbf{d}^0 = 0$ . In our experiments, we have used a value for  $\lambda$  of 0.5.



**Fig. 5.** Test Sequence 3, tracking with RGB histograms. a), b) and c): simple models shown in Figure 2; d): multiple model (on the upper-left corner of the images, the best model for that frame is shown).

Due to the simplicity of the dynamical model, we have used a  $\sigma$  value in Equation (7) of 0.5. Having a more precise and less general dynamics would allow this value to be reduced, favoring measurements closer to the expected position of the target.

Although there are no significant lighting variations in the first sequence (Figure 3), the multiple model performs better when the hand and the ball occlude the target, because the dynamics gives less weight to these distracting events. The results obtained with RGB and YUV-based models are very similar.

In the second and third sequence, significant variations can be observed in the lighting conditions of the target. The simple model obtained from Image C (Figure 2) gets distracted at the beginning of the second sequence (Figure 4) because of the similarity between the histograms of the head and the ground. For the last sequence, we can see that the selection of the best model for each frame increases the tracker accuracy (Figure 5).

## 4 Conclusion

The method presented in this paper enables multiple models to be used in order to prevent loss when there are significant variations in the target's histogram, and allows real-time execution on a desktop computer.

The experimental results indicate that our method increases the robustness of tracking when faced with lighting changes in the object. By adequately selecting the samples for the multiple model, it is possible to track an object from its generic set of images.

## Acknowledgment

This work has been financed by grant TIC-2001-3316 from the Spanish Ministry of Science and Technology.

## References

1. M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proceedings of European Conference on Computer Vision*, Cambridge, UK, 1996, pp. 343–356.
2. Hedvig Sidenbladh and Michael J. Black, "Learning image statistics for bayesian tracking," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Vancouver, Canada, 2001, vol. 2, pp. 709–716.
3. Y. Wu and T.S. Huang, "A co-inference approach to robust visual tracking," in *Proceedings of Eighth IEEE International Conference on Computer Vision*, 2001, vol. 2, pp. 26–33.
4. S. Avidan, "Support vector tracking," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2001, pp. 184–191.
5. D. Gavrila and V. Philomin, "Real-time object detection for smart vehicles," in *Proceedings of IEEE International Conference on Computer Vision*, 1999, pp. 87–93.
6. K. Toyama and A. Blake, "Probabilistic tracking in a metric space," in *Proceedings of IEEE International Conference on Computer Vision*, 2001, vol. 2, pp. 50–57.
7. G.R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, 1998.
8. Tyng-Luh Liu and Hwann-Tzong Chen, "Real-time tracking using trust-region methods.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 397–402, 2004.
9. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.
10. T. Kailath, "The divergence and bhatacharyya distance measures in signal selection," *IEEE Transactions on Comm. Technology*, vol. 15, pp. 52–60, 1967.



# Efficient Object-Class Recognition by Boosting Contextual Information

Jaume Amores<sup>1,\*</sup>, Nicu Sebe<sup>2</sup>, and Petia Radeva<sup>1</sup>

<sup>1</sup> Computer Vision Center, Universitat Autònoma de Barcelona

<sup>2</sup> University of Amsterdam

**Abstract.** Object-class recognition is one of the most challenging fields of pattern recognition and computer vision. Currently, most authors represent an object as a collection of parts and their mutual spatial relations. Therefore, two types of information are extracted: local information describing each part, and contextual information describing the (spatial) context of the part, i.e. the spatial relations between the rest of the parts and the current one. We define a generalized correlogram descriptor and represent the object as a constellation of such generalized correlograms. Using this representation, both local and contextual information are gathered into the same feature space. We take advantage of this representation in the learning stage, by using a feature selection with boosting that learns both types of information simultaneously and very efficiently. Simultaneously learning both types of information proves to be a faster approach than dealing with them separately. Our method is compared with state-of-the-art object-class recognition systems by evaluating both the accuracy and the cost of the methods.

## 1 Introduction

In this work we deal with the problem of detecting the presence or absence of one object category in an image. In contrast to simple object recognition, object-class recognition is not restricted to images of the same physical object (e.g. different images of the same car), but deals with different instances of the object, e.g. images of different cars. This introduces a high variability of appearance across objects of our category. The difficulty is increased by the presence of clutter in the images, partial occlusion and accidental conditions in the imaging process. Among recent approaches, characterizing the object as a collection of parts and their spatial arrangement has proved to be a promising direction [1–4].

Classical contextual representations such as Attribute Relational Graph (ARG) [4] and constellation of parts [3, 5] deal separately with these two forms of information: local information is represented by feature vectors associated to each part and contextual information is represented by a set of relative spatial vectors, i.e. differences in spatial position.

In this work we define a constellation of generalized correlograms for object representation. Correlograms were used to measure the joint distribution of pixel-level color information along with the spatial distribution [6]. A generalized correlogram is introduced here to deal with higher level properties related to parts of an object. The image

---

\* Work supported by CICYT TIC2000-1635-C04-04, Spain.

is represented by constellations of such generalized correlograms, instead of using a unique descriptor as done originally [6]. Belongie et al. [7] introduced constellations of shape contexts (another type of correlogram) that only deal with spatial arrangements, i.e. do not consider local information. In our representation, every feature vector in our feature space gathers local and contextual information. A great advantage of this feature space is that it can be integrated with an efficient feature selection and learning algorithm such as AdaBoost [8, 9] with weak classifiers based on single dimensions. This leads to simultaneously learning those spatial relations and local properties of parts that are characteristic of the object category.

Summarizing, the main contribution of this work is in integrating a new constellation of generalized correlogram representation into AdaBoost with feature selection. AdaBoost used with weak classifiers based on single dimensions together with our object representation lead to an efficient object recognition scheme dealing with the spatial pattern of the object. We first explain the image representation in Section 2, followed by the description of the spatial pattern classifier with boosting in Section 3. In Section 4 we report results and conclude in Section 5.

## 2 Image Representation

In this section we introduce a new representation of the object by using a constellation of generalized correlograms. Let an image  $I_k$  be represented by a constellation of  $U_k$  object parts, expressed as  $H_k = \{\langle o_i, \mathbf{h}_i, \mathbf{x}_i \rangle\}_{i=1}^{U_k}$ . The  $i$ -th detected part is represented by the tuple  $\langle o_i, \mathbf{h}_i, \mathbf{x}_i \rangle$ , where  $o_i$  is the label identifying the part,  $\mathbf{h}_i$  are the properties describing the part, and  $\mathbf{x}_i$  is its spatial position in the image. Due to clutter, parts in  $H_k$  might correspond to different objects. Let  $X_k = \{\mathbf{x}_i\}_{i=1}^{U_k}$  be the set of spatial positions of parts from  $H_k$ . One way to obtain potential parts of an image is by extraction of interesting points, also called features or key points [3, 5], this is also our approach.

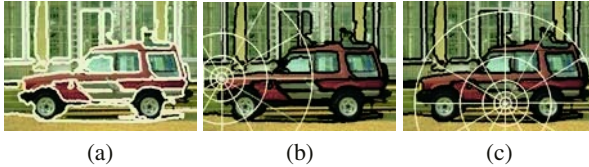
For our purpose, it is important to not miss any informative location, and to perform a fast interesting point extraction. By interesting point we mean any point located at an informative position, such as the edges, we do not mean necessarily corners. Two levels of interesting points are extracted. First we obtain a dense set of interesting points representing potential parts of objects. From this dense set, we extract local information around each point. Let  $H_k^L = \{\langle o_i^L, \mathbf{h}_i^L, \mathbf{x}_i^L \rangle\}_{i=1}^{U_k^L}$ ,  $\mathbf{x}_i^L$  denote this dense set (do not confuse with the final representation  $H_k$ ). We extract local information as properties  $\mathbf{h}_i^L$  of these parts. Let  $X_k^L = \{\mathbf{x}_i^L\}_{i=1}^{U_k^L}$  be the dense set of positions from  $H_k^L$ . In our implementation, these positions are located at extracted contours of the image, see fig. 1(a). From  $X_k^L$  we sample a much more sparse set of interesting points  $X_k \subset X_k^L$  covering the different locations from which we measure the relative spatial distribution of local properties in  $H^L$ .  $X_k$  contain the positions of our final constellation  $H_k$ . Each point  $\mathbf{x}_j \in X_k$  is the position of  $o_j$ . We associate as descriptor  $\mathbf{h}_j$  a correlogram that measures the joint distribution of spatial relations  $(\mathbf{x}_i^L - \mathbf{x}_j)$  and local properties  $\{\mathbf{h}_i^L\}_{i=1}^{U_k^L}$ . Let us express the spatial relation  $(\mathbf{x}_i^L - \mathbf{x}_j)$  in polar coordinates:  $(\alpha_{ij}, r_{ij})$ , and the  $d$  local properties as  $\mathbf{h}_i^L = (l_{i1}, l_{i2}, \dots, l_{id})$ . The joint distribution is measured by a histogram based on a partition of the  $d + 2$  dimensional space with vectors  $\mathbf{v}_{ij} =$

$(\alpha_{ij}, r_{ij}, l_{i1}, l_{i2}, \dots, l_{id}), i = 1, \dots, U_k^L$ . The partition of this space is obtained by intersection of separate partitions made for each individual dimension. Let  $B_w$  be the  $w$ -th bin in the final  $d + 2$  space. The  $j$ -th correlogram is expressed as  $h_j(w) = \frac{1}{U_k^L} |\{v_{ij} \in B_w, i = 1, \dots, U_k^L\}|$ , i.e. the  $w$ -th bin of  $h_j$  counts the number of vectors  $v_{ij}$  falling into this bin.

Note that this space contains vectors that express spatial relations and local properties, and thus the resulting descriptor  $h_j$  is a correlogram of local properties in  $H_k^L$  considering their spatial distribution around the point of reference  $x_j$ . As  $X_k \subset X_k^L$ , we are describing in the same vector  $h_j$  attached to  $o_j$  the local properties of  $o_j$ , the local properties of the rest of parts in the dense set  $H^L$ , and the spatial distribution of these parts relative to  $o_j$ .

The dense set of interesting points in  $X_k^L$  is obtained by extracting the contours from an over-segmentation with k-means and subsequent postprocessing that obtains spatially contiguous blobs. The sparse set of points in  $X_k$  is sampled from  $X_k^L$  keeping points with maximum spatial distance to each other, so that  $X_k$  covers points of view from different angles of the image (see fig. 1(b)-(c)). An important characteristic of our implementation is that it is fast, and the results show that allows accurate representation.

For the spatial dimensions, we use the same log-polar spatial quantization as the shape-context correlogram of Belongie et al [7] (see fig. 1(b)-(d)). This makes the descriptor  $h_j$  focus more on local properties around  $o_j$  (local context) than to the far context. The dimensions regarding the local properties  $l_{i1}, l_{i2}, \dots, l_{id}$  are linearly quantized; we explain below each of them in turn.



**Fig. 1.** (a) Dense cloud of points covering interesting parts of the image (edges). (b)-(c) Log-polar spatial quantization of our correlogram. Each descriptor in (b) and (c) represents a different “point of view” of the object’s spatial arrangement

As local information, local structure and color around a small neighborhood are utilized. As local structure, the local direction of the edges is used. Specifically, the angle is measured along the curve formed by contours. After smoothing the contours, the angle is taken modulus  $\pi$ , and we make a quantization into 4 bins. The color is linearly quantized and mapped into one dimension. We perform a very coarse quantization of the R,G,B space into 3, 2, 2 bins to avoid large feature vectors in the final histogram. As there is not only one dominant color around the local part  $o_i^L$ , we take every color around a small neighborhood and consider the proportion of this color in this neighborhood, thus a local color histogram is taken. In this way, we are performing a fuzzy assignment of the part  $o_i^L$  to bins of the (local) color space, using the local color histogram  $h_i^c : \{1, 2, \dots, 12\} \rightarrow [0, 1]$  as the *color membership function* of  $o_i^L$ .

Different authors have used correlograms [6, 7]. The common feature is to use pixel-level properties, traditionally only color, considering every pixel in the image. High-

level entities such parts of objects are not considered in their formulation. Authors do not consider constellations of their correlograms but aggregate all the descriptors into one single (spatial) histogram for the image. Belongie et al. [7] use constellations of shape contexts but do not use any local information, they describe binary contours by the presence of a spatial position. The definition presented here can be considered a generalization of correlograms into a constellation of parts framework. One drawback of the spatial quantization we use is that it must be scaled with the size of the object to provide scale invariance. This scaling is done by normalizing the distances  $r_{ij}$  by the size of the object. As we do not know a priori the size of our objects, we must compute the contextual descriptors for different scales fixed a priori. Let  $n_s$  be the number of scales (experimentally we chose  $n_s = 7$ ). The final representation of the image  $I_k$  is expressed as  $A_k = \{H_k^s\}_{s=1}^{n_s}$ , where  $H_k^s$  is the set of parts of  $I_k$  with contextual descriptors  $h$  scaled according to scale  $s$ .

### 3 Learning Multiple Contextual Representations with Boosting

The explained representation is suitable for combination with a feature selection and learning method such as AdaBoost with weak classifiers based on single dimensions, that proved to be very efficient [8, 9]. By learning the relevant dimensions of vectors  $\mathbf{h}$  defined in section 2, we are simultaneously learning the properties characterizing every part of the object and their mutual spatial relations.

In our framework, the model of one object is expressed as  $\Omega = \{\langle \omega_i, \varphi_i \rangle\}_{i=1}^M$ , where  $\omega_i$  is the label of one model part,  $\varphi_i$  are the parameters for this model part learnt by the classifier, and  $M$  is the number of model parts. We denote as  $l_i^\omega(o_j | o_j \in H_k^s)$  the likelihood that part  $o_j \in H_k^s$  from image  $I_k$  with scale  $s$  represents the model part  $\omega_i$ . We denote as  $L_i^\omega(H_k^s)$  the likelihood that any  $o_i$  in  $I_k$  with scale  $s$  represents the model part  $\omega_i$ . As we are using contextual descriptors,  $\omega_i$  also represents the whole model object according to one particular point of view. Therefore,  $L_i^\omega$  conveys a piece of evidence of the existence of the model object according to the point of view  $\omega_i$ .  $L_i^\omega(H_k^s)$  is the likelihood that *any*  $o_j \in H_k^s$  represents  $\omega_i$ , we apply as OR rule the maximum so that  $L_i^\omega(H_k^s) = \max_{o_j \in H_k^s} l_i^\omega(o_j | o_j \in H_k^s)$ . This can also be regarded as matching  $\omega_i$  with some  $o_m \in H_k^s$ , which is expressed as  $M_i^\omega(H_k^s) = o_m = \arg \max_{o_j \in H_k^s} l_i^\omega(o_j | o_j \in H_k^s)$ .

Based on the individual likelihoods  $L_i^\omega$ , we denote as  $L^\Omega(H_k^s)$  the likelihood that the object exists in  $I_k$  with scale  $s$ , according to the whole model  $\Omega = \{\langle \omega_i, \varphi_i \rangle\}_{i=1}^M$ . As we want all the model points of view  $\omega_i$  of the object to contribute to this likelihood, we use as combination rule the mixture  $L^\Omega(H_k^s) = \sum_{i=1}^M \frac{1}{M} L_i^\omega(H_k^s)$ .

Recall that the image  $I_k$  is represented by different scales  $A_k = \{H_k^s\}_{s=1}^{n_s}$ . The likelihood that the object exists with *any* scale in the image representation  $A_k$  is expressed as  $L_f^\Omega(A_k) = \max_{H_k^s \in A_k} L^\Omega(H_k^s)$ , where we have applied again the maximum as OR rule. Again, this can be regarded as matching the model object with some *scaled* representation  $H_k^m$  in  $A_k$ , which we express as  $M_s(A_k) = H_k^m = \arg \max_{H_k^s \in A_k} L^\Omega(H_k^s)$ .

To learn the model  $\Omega$ , AdaBoost is applied over each separate model point of view  $\omega_i$ . This requires a separate training set for each  $\omega_i$ . We denote as  $T_i$  this training set.  $T_i$  contains as positive samples the parts  $o_j$  matching  $\omega_i$  with the correct scale in every

*positive* image (i.e. an image containing an object of the category we are learning). As negative samples  $T_i$  contains every part  $o_j$  with every scale in *negative* images. The problem of matching is solved in two stages. First, robust matchings are extracted from a small set of manually segmented images from the training set (we will see that very few images are enough). This is carried out by performing non-rigid registration [7] over these manually segmented images, which obtains an initial training set  $T'_i$  for each  $\omega_i$ .  $T'_i$  contains as positive instances only those from the segmented subset, but has many negative instances, as we use every part with every scale in every negative image. This allows to discard a lot of structures from clutter. We learn an initial model part  $\omega_i$  with  $T'_i$ . With the learnt model part, we can now match corresponding parts  $o_j$  with corresponding scales in the rest of images not segmented manually to construct the final big training set  $T_i$ . Registration is not robust in clutter, therefore we match  $\omega_i$  with those  $o_j$  that have high likelihood according to the previous learning. We apply in every positive image first the scale matching  $M_s(A_k)$  and then we apply the part matching in the appropriate scale  $M_i^\omega(M_s(A_k))$  (see the expressions above). Finally, we train again the model with the complete training set  $T_i$  and obtain the final classifier for the whole model object  $\Omega$ .

## 4 Results

We used the CALTECH database (<http://www.robots.ox.ac.uk/vgg/data3.html>) collected by Fergus et al. [3, 10], which consists of 7313 images with clutter for object recognition. Part of this database was also used by other authors such as Agarwal et al. in [1]. This database contains 7 different categories, which is a big step forward compared to many databases used in other works that are based on one or two categories. A full description of the database along with examples can be found in [3, 10], we do not show them here due to lack of space. The object categories can be found in table 1(a). Most of the object categories have instances under the same bi-dimensional arrangement, except for the spotted-cat category taken from Corel<sup>®</sup> database. Each category has roughly 800 images of different objects of this category. From the positive training set, 10 images are manually segmented. The negative set of images were taken by Fergus et al. from Google<sup>®</sup> by searching with the keyword “things”. This consists of 520 images, 400 were taken as training and 120 as test. Each time the training consisted of 400 positive images, 400 negative, and the test of 100 positive and 100 negative.

A cross-validation procedure was followed to test a total of 400 negative images and 400 positive images, average results are shown. Each time, the images included in the sets were picked randomly, always using disjoint sets for training and test.

Fig. 2(a)-(b) shows a example of results for 2 of the 7 categories, motorbikes in (a) and faces in (b). Fig. 2(a) shows every image correctly classified as motorbike. Some images show a heavy clutter and still there are no incorrect matches. Fig. 2(b) shows images classified as faces. Faces show an incorrect match, that can be seen to be similar in shape.

In fig. 3 each row shows the matching from a part  $\omega_i$  of the learnt model, to a matched part  $o_j$  in different instances of the object. In the first row we show matching of one model part of the car(rear) category. We can see that the model part is consistently



**Fig. 2.** (a) Example of images correctly classified as motorbikes, (b) images classified as faces



**Fig. 3.** Some matchings obtained with in several classes

matched with the same shadow beneath the car in the images. In the second row, another model part is matched consistently near the left red light in the images. In the motorbike category (third row), one model part matches with parts in a similar relative position of the instance motorbikes, despite the clutter. Finally, a model part of the face category (forth row) matches with parts near the ear of the face instances.

Given a test set with positive and negative images, the goal is to detect what images contain some instance of the object category and what do not contain any instance. The classification hit rate is measured using the receiver-operating characteristic (ROC) equal error rates:  $p(\text{True Positive})=1-p(\text{False positive})$ . Table 1(a), presents results comparing our method against the constellation used by Fergus et al. in [3], they also report results with other approaches using the same data set (see reference). In all the categories except the spotted cat and face, our method outperforms the one reported by Fergus et al. The spotted cat has very different poses which makes the spatial quantization that we use not so suitable. However, the inclusion of local properties such as color makes boosting focus more on local information than contextual information in this category, so that not bad results are obtained. The face category is probably better

**Table 1.** (a) ROC equal error rates measures with the method in [3], and our method (b) Computational cost for different stages, see text for explanation, the second row is the inverted file arrangement cost, only in our system

(a)			(b)		
Category	[3]	Ours	Step	[3]	Ours
Car(Rear)	90.3%	96.9%	Description per image	15 sec.	6 sec.
Plane	90.2%	94.5%	I.F. per image	-	2 sec.
Leaf	-	96.3%	Training	36 hours	4 hours
Motorbike	92.5%	95.0%	Classification per image	3 sec.	0.23 sec.
Face	96.4%	89.5%			
Spotted Cat	90.0%	86.5%			
Car (Side)	88.5 %	90.0%			

represented using local appearance and PCA as Fergus does, we can include this in a future work. For the car (side) category the result is a recall-precision equal error. The negative set in this category contain images of roads without cars [3], so that a more realistic experiment can be made.

To speed up the algorithm, we take the non-empty bins of correlograms describing the current object category, and only use these bins in AdaBoost. That is, bins that are empty in our positive training set  $T_i$  (see section 3) are not used by the classifier. This also makes the algorithm more robust against clutter because we disregard structures not found in our object category. A similar idea was used in [11] for shape contexts. We also make use of the high sparseness of our generalized correlograms both in the training and in the recognition stage. Note that a correlogram is a special type of histogram. We only process the non-zero elements of our descriptors, by structuring the data as inverted files, a technique used in information retrieval [12]. For each dimension we keep the index of the descriptors that contain a non-zero value for this dimension, along with the value for this descriptor. Then the descriptors are sorted by the value of this dimension. This allows to use binary search in the recognition stage when we are looking for values in one dimension exceeding the threshold obtained by AdaBoost [9]. As our images contain a large constellation of descriptors with different scales (typically 100 descriptors with 7 scales) this technique speeds up the algorithm by obtaining a logarithmic cost in the number of scanned descriptors. Although sorting has a cost a bit higher than linear, it is done only once, saving later a lot of cost in the search for each model part  $\omega_i$ . Furthermore, sorting has linear cost on the number of descriptors that have a value greater than zero, only 20% of the descriptors due to the sparseness. This technique is also suitable for retrieving in large databases if we have pre-computed the descriptors. The time cost for every stage is shown in table 1(b), compared to the method in [3]. The second row denotes the cost of arranging the description of  $I_k$  as inverted file and sorting. We used a computer at 2.4 GHz for the experiments, while experiments in [3] were made with a computer at 2.0 GHz. We used Matlab<sup>®</sup> with some subroutines in C. Some parts such as the feature extraction and training stage could be made more efficient.

## 5 Discussion

In this work a an object class recognition system has been proposed that is able to learn the characteristic parts of the object and their spatial relationship in the presence of clutter. The image is represented as a constellation of very sparse contextual descriptors and this representation is integrated with an efficient feature selection and learning algorithm such as boosting. We achieved very accurate classifier compared to the approach of Fergus et al. [3]. Furthermore, making use of the sparseness we showed that an efficient method can be achieved, suitable for scanning large databases. Summarizing, our novel contribution is to propose an efficient object class recognition framework that incorporates a novel constellation of contextual descriptors into an efficient boosting algorithm used with feature selection.

For future research, we would like to enrich the feature space by combining the log-polar spatial quantization with other types of spatial quantization less sensitive to shape, in order to be able to recognize the same object under different spatial configurations (for example a dog with different poses). By boosting we can combine a descriptor sensitive to different shapes and a (contextual) descriptor robust against shape variations, and let the classifier learn if the object is very structured.

## References

1. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. *IEEE TPAMI* **26** (2004) 1475–1490
2. Schneiderman, H.: Learning a restricted bayesian network for object detection. In: *IEEE Proc. CVPR*. (2004) 639–646
3. R., F., P., P., A., Z.: Object class recognition by unsupervised scale-invariant learning. In: *IEEE Proc. CVPR*. (2003)
4. Hong, P., Huang, T.S.: Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs. *Journal of Discrete Applied Mathematics* **139** (2003) 113–135
5. Weber, M., Welling, M., Perona, P.: Towards automatic discovery of object categories. In: *IEEE Proc. CVPR*. (2000) 101–108
6. Huang, J., Kumar, S., Mitra, M., Zhu, W., Zabih, R.: Image indexing using color correlograms. In: *IEEE Proc. CVPR*. (1997) 762–768
7. Belongie, S., Malik, J., J.Puzicha: Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI* **24** (2002) 509–522
8. Schapire, R.E., Singer, Y.: Improved boosting using confidence-rated predictions. *Machine Learning* **37** (1999) 297–336
9. Viola, P., Jones, M.J.: Robust-real time face detection. *Int'l J. of Computer Vision* **57** (2004) 137–154
10. Fei-Fei, L., Fergus, R., Perona, P.: A bayesian approach to unsupervised one-shot learning of object categories. In: *IEEE Proc. ICCV*. Volume 2. (2003) 1134–1142
11. Thayananthan, A., Stenger, B., Torr, P., Cipolla, R.: Shape context and chamfer matching in cluttered scenes. In: *IEEE Proc. CVPR*. (2003)
12. Squire, M.D., Muller, H., Muller, W.: Improving response time by search pruning in a content-based image retrieval system, using inverted file techniques. In: *IEEE Workshop CBAIVL*. (1999)



# Illumination Intensity, Object Geometry and Highlights Invariance in Multispectral Imaging<sup>\*</sup>

Raúl Montoliu<sup>1</sup>, Filiberto Pla<sup>2</sup>, and Arnaud C. Klaren<sup>2</sup>

<sup>1</sup> Dept. Arquitectura y Ciencias de los Computadores, Jaume I University  
Campus Riu Sec s/n 12071 Castellón, Spain

[montoliu@uji.es](mailto:montoliu@uji.es)

<sup>2</sup> Dept. Lenguajes y Sistemas Informáticos, Jaume I University  
Campus Riu Sec s/n 12071 Castellón, Spain

[{pla,klaren}@uji.es](mailto:{pla,klaren}@uji.es)

<http://www.vision.uji.es>

**Abstract.** It is well-known that image pixel values of an object could vary if the lighting conditions change. Some common factors that produce changes in the pixels values are due to the viewing and the illumination direction, the surface orientation and the type of surface.

For the last years, different works have addressed that problem, proposing invariant representations to the previous factors for colour images, mainly to shadows and highlights. However, there is a lack of studies about invariant representations for multispectral images, mainly in the case of invariants to highlights.

In this paper, a new invariant representation to illumination intensity, object geometry and highlights for multispectral images is presented. The dichromatic reflection model is used as physical model of the colour formation process. Experiments with real images are also presented to show the performance of our approach.

## 1 Introduction

The image pixel values of an object could vary if the lighting conditions change. During the image formation process, the main factors that could produce changes in the pixel values are: viewing direction, surface orientation, highlights, illumination direction, illumination intensity, illumination colour and inter-reflections.

The aim of invariant image representations is to obtain the same value for the pixels of an object, independently of the conditions commented above. These representations can be quite useful to measure or recognize objects in images or other tasks that require invariance to any of these properties. For instance, intensity-based edge detectors cannot distinguish the physical cause of an edge, such as material, shadows, surface orientation changes, etc. This fact produces poor segmentations and, therefore, bad recognition of objects.

---

<sup>\*</sup> This paper has been partially supported by projects: DPI2001-2956-C02-02 from Spanish CICYT and IST-2001-37306 from European Union.

For the last years, significant works about invariant representations for colour images have been carried out [2], [4], [1]. Many of them use the reflection model introduced by Shafer in [7] as a physical model to understand the colour of a concrete pixel. The reader is addressed to [3] for a comprehensive study.

The next section explains how to obtain invariant representations to illumination intensity and other geometric factors (as shadows) and highlights, performing simple mathematical operation with bands (R, G and B, for colour images). Our approach for multispectral images is based on similar properties, taking advantage of the Neutral Interface Reflection (NIR) and narrow band filter assumptions. We have named our invariant  $L_n$  which is invariant to illumination intensity (assuming white illumination), object geometry and highlights while approximately preserving the spectral information of the image.

## 2 Multispectral Invariant Representations

The use of the reflection model is key point to understand how a sensor works. The Dichromatic reflection model introduced by [7], represents the output value  $C$  of a pixel in the image plane as:

$$C_n = m_b(\vec{n}, \vec{s}) \int_{\lambda} f_n(\lambda) e(\lambda) c_b(\lambda) d\lambda + m_s(\vec{n}, \vec{s}, \vec{v}) \int_{\lambda} f_n(\lambda) e(\lambda) c_s(\lambda) d\lambda \quad (1)$$

for  $C_n = \{C_1, \dots, C_N\}$  giving the  $C_{th}$  sensor response of a multispectral camera,  $c_b$  and  $c_s$  are the surface albedo and Fresnel reflectance respectively,  $\lambda$  denotes the wavelength,  $\vec{n}$  is the surface patch normal,  $\vec{s}$  is the direction of the illumination source and  $\vec{v}$  is the direction of the viewer. Geometric terms  $m_b$  and  $m_s$  denote the geometric dependencies on the body and surface reflection component respectively.

Considering the Neutral Interface Reflection (NIR) model (assuming that  $c_s(\lambda)$  has a constant value independent of the wavelength), narrow band filters modelled as a unit impulse and white illumination (equal energy density for all wavelengths within the visible spectrum), then  $e(\lambda) = e$ ,  $f = \int_{\lambda} f_1(\lambda) d\lambda = \dots = \int_{\lambda} f_N(\lambda) d\lambda$  and  $c_s(\lambda) = c_s$ , and hence being constants. Then, with this assumption, the measured sensor values are given by:

$$C_n = em_b(\vec{n}, \vec{s})K_n + em_s(\vec{n}, \vec{s}, \vec{v})c_s f \quad (2)$$

with  $K_n = \int_{\lambda} f_n(\lambda) c_b(\lambda) d\lambda$ .

If the object is matte, that is, if it does not have highlights, then the second part of the equation 2 can be neglected. Therefore, the equation 2 can be simplified as follows:

$$C_n = em_b(\vec{n}, \vec{s})K_n \quad (3)$$

It is possible to obtain invariant representations to some conditions, performing simple mathematic operations with the bands. For instance: for matte objects, dividing two bands  $i, j$  allows to get an illumination intensity and object geometry invariant representation, i.e. non-dependent of  $m_b$  and  $e$  factors:

$$\frac{C_i}{C_j} = \frac{em_b(\vec{n}, \vec{s})K_i}{em_b(\vec{n}, \vec{s})K_j} = \frac{K_i}{K_j} \quad (4)$$

For shiny objects, subtracting one band from another provides a highlights invariant representation, i.e. invariant to viewpoint  $m_s$  and specular reflection coefficient  $c_s$ :

$$\begin{aligned} C_i - C_j &= (em_b(\vec{n}, \vec{s})K_i + em_s(\vec{n}, \vec{s}, \vec{v})c_s f) \\ &\quad - (em_b(\vec{n}, \vec{s})K_j + em_s(\vec{n}, \vec{s}, \vec{v})c_s f) \\ &= em_b(\vec{n}, \vec{s})(K_i - K_j) \end{aligned} \quad (5)$$

Finally, first subtracting and then dividing bands provides a representation invariant to highlights, illumination intensity and object geometry:

$$\frac{C_i - C_j}{C_k - C_l} = \frac{em_b(\vec{n}, \vec{s})(K_i - K_j)}{em_b(\vec{n}, \vec{s})(K_k - K_l)} = \frac{K_i - K_j}{K_k - K_l} \quad (6)$$

Following these ideas, Stockman and Gevers [8] presented two invariant representation for multispectral images, the normalized hyper-spectra and the hyper-spectral hue.

The normalized hyper-spectra is a representation invariant to  $e$  and  $m_b$  factor. It is defined as follows:

$$c_n = \frac{C_n}{C_1 + \dots + C_N} \quad (7)$$

The calculation of the hyper-spectral hue needs a special attention since hue orders colors in a circular way. First an equal-energy illumination is obtained dividing each band by the corresponding sensor response of a white reference object, and supposing that the filter is a narrow band filter modelled as a unit impulse [8]. Thus, the object can be made independent of the illumination intensity.

In a second step, all the values are transformed as follows:

$$c_n = C_n - \min(C_1 + \dots + C_N) \quad (8)$$

As a result, the transformed spectrum is invariant to highlights.

After the pre-processing of the spectrum, the hue can be calculated using the following equation:

$$H(c_1, \dots, c_N) = \arctan \left( \frac{\sum_i c_i \cos(\alpha_i)}{\sum_i c_i \sin(\alpha_i)} \right), \text{ where } \alpha_i = \frac{(i-1)2\pi}{N} \quad (9)$$

As a result, the transformed spectrum is also invariant to object geometry. The reader is addressed to [8] for further details.

### 3 $L_n$ Multispectral Invariant

The multispectral Hue is invariant to illumination intensity (assuming white illumination), object geometry and highlights which are the properties that we

are looking for. Nevertheless, the fact that it transforms an image with  $N$  bands to an image with just 1 band can produce an import loose of multispectral information, which can be crucial in many applications.

Therefore, we propose the  $L_n$  invariant for multispectral images which transforms an image with  $N$  bands into an invariant representation with  $N - 2$  independent bands. It is defined as follows:

$$L_n = \frac{C_n - \min(C_1, \dots, C_N)}{\sum_j (C_j - \min(C_1, \dots, C_N))} \quad (10)$$

In order to make the acquired images independent from illumination, the aperture times of our multispectral camera have been calculated carefully for every band to eliminate differences in light intensity that are caused by the spectrum characteristics of the lamps, the filter and the sensor. This calculation is done by repeatedly taking multispectral images of a white reference, (i.e. a white surface with equal reflection properties in a wide spectrum) and adjusting the aperture times until the light intensity is the same in every band. This process is called white balancing. These aperture times compensate for the unknown spectral characteristics of the lamps, the filter and the sensor. Thanks to that process, we can assume that we are using white illumination and therefore the acquired images fulfill that  $e(\lambda) = e, \forall \lambda$ . This fact allows to suppose that the sensor behaviors following Equation 2.

The aim is to obtain an invariant representation where the spectral information is preserved, i.e. the invariant pixel value not to be a mixture of other pixel (wavelengths) values. Lets,  $C_i = em_b K_i + B$ ,  $C_j = em_b K_j + B$  and  $\min(C_1, \dots, C_N) = C_{min} = em_b K_{min} + B$ , with  $B = em_s c_s f$  being a constant value along  $\lambda$ ,  $m_b = m_b(\vec{n}, \vec{s})$  and  $m_s = m_s(\vec{n}, \vec{s}, \vec{v})$ . In order to achieve highlights invariance, we can perform  $C_i - C_j$ , but then, a mixture of body reflectance values from both pixels is obtained as an invariant, loosing spectral information,  $C_i - C_j = em_b (K_i - K_j)$ . However, using the minimum value, the spectral information is approximately preserved, since  $C_{min} = em_b K_{min} + B \simeq B$  and therefore  $C_i - C_{min} \simeq em_b K_i + B - B = em_b K_i$ , i.e. invariant to highlights. In addition,  $L_n$  is also invariant to  $e$  and  $m_b$ , i.e. illumination intensity and geometry factors, since:

$$L_n \simeq \frac{em_b K_n}{\sum_j em_b K_j} = \frac{K_n}{\sum_j K_j} \quad (11)$$

Note that we are dividing all the pixel values by a constant, therefore the spectral information is maintained.

## 4 Experimental Results

In order to test our approach in real images, a set of multispectral images have been taken using a specially designed illumination chamber (see Figure 1). The chamber is a perfect hemisphere with a large number of low-voltage halogen lamps attached on the inside uniformly distributed through the hemisphere. The

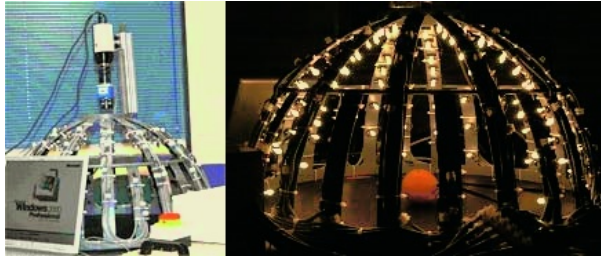


Fig. 1. Illumination chamber used to capture our multispectral images

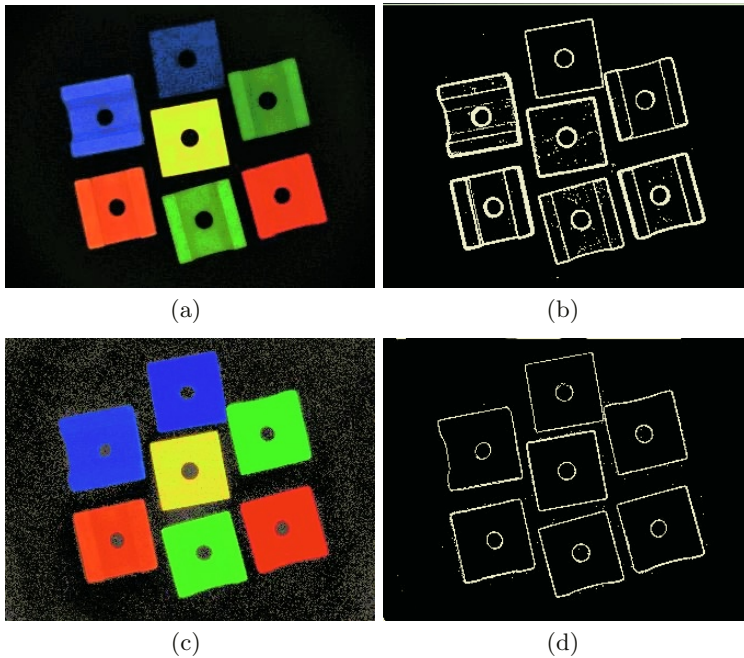
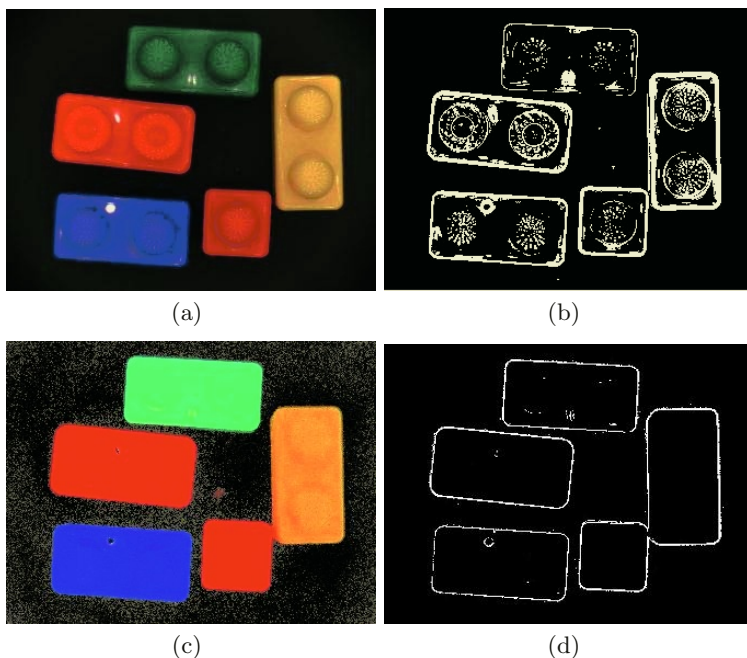


Fig. 2. Wooden toys experiment. (a) original image, (b) original edge-image, (c)  $L_n$  invariant representation, (d)  $L_n$  invariant edge-image. See text for explanation

lamps illuminate the object from all sides and from equal distances, minimizing shadows, shine and other effects. For each image, 33 bands have been captured, from  $400nm$  to  $720nm$ , using a bandwidth of  $10nm$ .

From the experiments performed using the set of images captured, the most significant ones are reported in this paper. Children toys have been selected as test objects since they have interesting properties that help us to demonstrate the invariant behaviour of our approach.

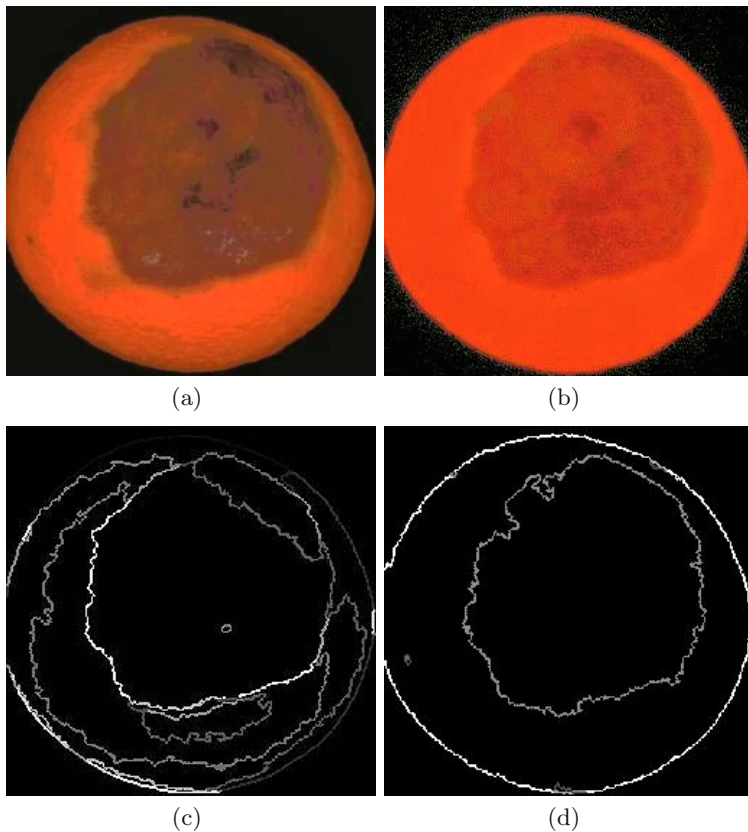
Figure 2 shows the "wooden toys" experiment. In figure 2a, the original 33-bands image is presented. In order to show the image as a RGB image, the bands  $650nm$ ,  $540nm$ ,  $490nm$  have been selected to be the R, G and B chan-



**Fig. 3.** Plastic Toys experiment. (a) original image, (b) original edge-image, (c)  $L_n$  invariant representation, (d)  $L_n$  invariant edge-image. See text for explanation

nels, respectively. Figure 2b shows the edge-image obtained from the 33 band original image. White pixels are the ones that are greater than a threshold in the multispectral gradient of the image. The gradient of the multispectral image has been calculated using the *Di Zenzo* multispectral gradient [9]. Note the edges produced by shadows in the objects. Figure 2c and 2d show the results of our approach. Figure 2c shows the  $L_n$  invariant representation as a RGB image ( $R = 650nm$ ,  $G = 540nm$  and  $B = 490nm$ ). Finally, Figure 2d shows the edge image obtained from the transformed multispectral image. Note that the effect of the shadows has been completely eliminated.

The next experiment involves plastic toys whose reflection properties produce highlights, which are hard to remove. Figure 3a shows the original image. As in the previous experiment, the bands  $650nm$ ,  $540nm$ ,  $490nm$  have been selected as the R, G and B channels, respectively. Figure 3b shows the edge-image obtained from the 33 band original image. Note the edges produced by shadows and highlights. Figure 3c shows the results of our invariant as a RGB image ( $R = 650nm$ ,  $G = 540nm$  and  $B = 490nm$ ). Finally, 3d shows the edge image obtained from the invariant image. Note that the effect of the shadows has been completely eliminated and the effect of the highlights has been almost completely eliminated. The brightest points have not been suppressed because of sensor saturation at these pixels.



**Fig. 4.** Orange segmentation experiment. (a) original image, (b)  $L_n$  invariant representation, (c) segmentation results of original image, (d) segmentation results of the transformed (by the invariant) image

In last experiment, our approach has been tested in an application to segment orange fruits. Figure 4a shows the original image as a RGB ( $R = 650nm$ ,  $G = 540nm$  and  $B = 490nm$ ). In spite of our effort to make an illumination chamber with a homogeneous illumination, the image of the orange shows variable illumination in different areas of the orange, higher in the center than in the periphery. Figure 4b shows the invariant representation, note that the illumination problems have been drastically reduced. In order to test if the invariant representation improves the segmentation of the orange, a multispectral segmentation algorithm has been used (see [6], [5]), using as input the original (Figure 4a) and the transformed image (Figure 4b). Figure 4c and 4d show both results.

Note the poor results of the segmentation using the original image due to the problems with illumination effects. On the other hand, note the excellent results of the segmentation process in Figure 4d, where the effect of the illumination problems have not influenced the extraction of the regions of the orange.

## 5 Conclusions

A new invariant for multispectral images has been presented in this paper. Our approach transforms the image into a new space which is invariant to illumination intensity (assuming white illumination), object geometry and highlights while approximately preserving the spectral information of the image.

The presented method has been successfully tested in real multispectral images with shadows and strong highlights, where it has been demonstrated the ability of the invariant to deal with those effects in the image and, therefore, can be used as input of other image processing applications, for instance, segmentation.

## References

1. Jan-Mark Geusebroek, Rein van Boomgard, Arnold W.M. Smeulders, and Hugo Geerts. Color invariance. *PAMI*, 23:1338–1350, 2001.
2. Th. Gevers and A. W. M. Smeulders. Color based object recognition. *Pattern Recognition*, 32:453–464, March 1999.
3. G. J. Klunker, S. A. Shafer, and T. Kanade. A physical approach to color image understanding. *International Journal of Computer Vision*, 4:7–38, 1990.
4. J.A. Marchant and C.M. Onyango. Shadow invariant classification for scenes illuminated by daylight. *Journal of the Optical Society of America A*, 2000.
5. A. Martinez-Usó, F. Pla, and P. Garcia-Sevilla. Multispectral segmentation by energy minimization. In *2nd Iberian Conference on Pattern Recognition and Image Analysis. IbPria'05*.
6. A. Martinez-Usó, F. Pla, and P. Garcia-Sevilla. Color image segmentation using energy minimization on a quadtree representation. In *International Conference on Image Analysis and Recognition, ICIAR'04*, 2004.
7. S. A. Shafer. Using color to separate reflection components. *Color Resolution Applications*, 10(4):210–218, 1985.
8. H. M. G. Stokman and Th. Gevers. Hyperspectral edge detection and classification. In *BMVC*, 1999.
9. S. Di Zeno. A note on the gradient of a multi-image. *Computer Vision Graphics Image Processing*, 33:116–125, 1986.



# Local Single-Patch Features for Pose Estimation Using the Log-Polar Transform

Fredrik Viksten and Anders Moe

Linköping University, Computer Vision Laboratory  
S-581 83 Linköping, Sweden  
{viksten,moe}@isy.liu.se

**Abstract.** This paper presents a local image feature, based on the log-polar transform which renders it invariant to orientation and scale variations. It is shown that this feature can be used for pose estimation of 3D objects with unknown pose, with cluttered background and with occlusion. The proposed method is compared to a previously published one and the new feature is found to be about as good or better as the old one for this task.

## 1 Introduction

Finding the geometrical state of an object from a single 2D image is of major importance for a lot of future applications in industrial automation such as bin picking and expert systems for augmented reality as well as a whole range of consumer products including toys and house-hold appliances. Previous research in this field has showed that there are a number of steps that need to fulfill a minimum level of functionality to make the whole system operational all the way from image to pose estimate. Important properties of a real-world system for pose estimation is robustness against changes in scale, lighting condition and occlusion. Robustness to scale is usually solved by some kind scale-space approach [9], but there are so far no really good ways to achieve robustness to lighting changes and occlusion. Occlusion is usually handled by using local features which is done here also. The local feature and the framework for pose estimation presented here has been tested in a setting that is constrained to the case of knowing what object to look for, but with no information on the state of the object. The inspiration to the work presented here comes from active vision and the idea of using steerable sensors with a foveal sampling around each point of interest [11]. Each point of interest detected in this work can be seen as a point of fixation for a steerable camera that then uses foveal sampling as a means of concentrating processing in the area close to that point.

### 1.1 Related Research

The problem of estimating object state has been investigated for as long as automated image processing has been possible. In the early period of the research

field, a lot of effort was spent on global methods, many without much real-world success. The work we will present here was in part inspired by one of those global methods [1]. In the recent years some advances have been made in the area of pose estimation [12], [10], [6], [8], of which much seems related to a new focus on local invariant features. Each local feature detected in an image during training can in such a setup be viewed as a search key to find the same view again from the database of learned object views.

## 1.2 Thesis of This Paper

We propose to use as local feature a patch of either the image or a edge-filtered version of the image and to use this feature in combination with a voting and clustering setup. An edge-filtered patch of the image can either be represented by the absolute value of the edge-vector in each point, or be represented in single or double angle notation [5]. The double angle representation effectively doubles the rotation angle around the z-axis for the edge normal in the image plane and thus has the advantage of not discriminating between lines or edges or the phase of an edge. This gives patches in double angle representation the chance to be more robust to changes in background and lighting. In this paper we will evaluate the performance of single and double angle represented patches.

We further propose that the patches are resampled with log-polar sampling and then transformed with the Fourier transform. This will in theory give us a local feature that can be made invariant to position in the image, to rotation and scale. When using discretized versions of continuous transforms like the log-polar transform used here, one has to be careful of how the discretization changes the transform, but we will show that this works in practice and is applicable to real-world setups for pose estimation.

## 2 Pose Estimation

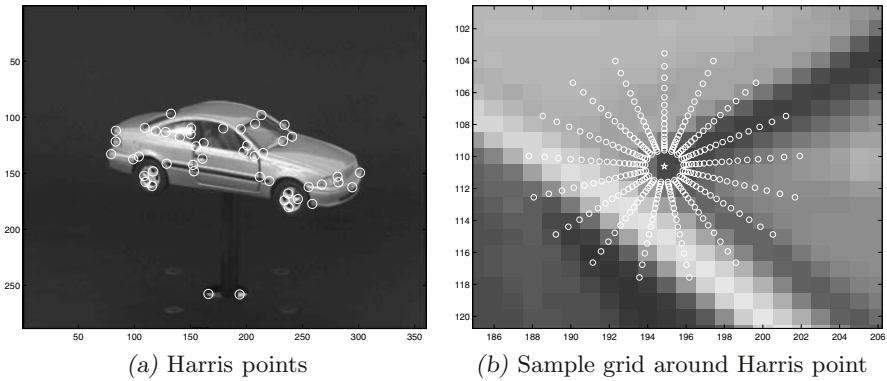
In this section we will describe the details behind our setup. What we will not go into detail about is the Harris corner<sup>1</sup> detector [7] that we use for feature selection. It was used since it seems to be one of the fastest and most stable ones around according to [13].

### 2.1 The Local Feature

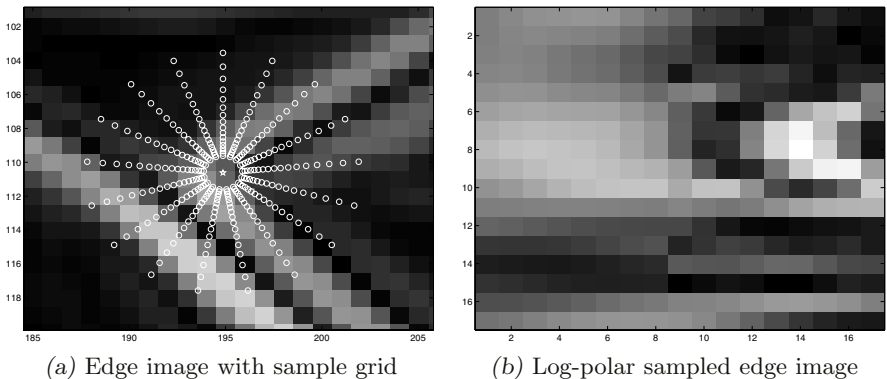
We see a zoom-in on one input image from the training set where Harris points have been drawn as small circles in Figure 1(a). Around each Harris point a log-polar sampling grid is placed in either the gray-valued image, see Figure 1(b), or an edge filtered image, see Figure 2(a). Resampling using this sampling grid and cubic interpolation yields an approximation to the log-polar transform for that local neighborhood, see Figure 2(b). In a log-polar sampled image, translation

---

<sup>1</sup> It is perhaps better to say that it detects non-simple signals.



**Fig. 1.** Harris points and log-polar sampling grid



**Fig. 2.** Edge image with sample grid and log-polar sampled edge image

equals rotation or scaling in the original image. It is possible to make this patch invariant to rotation and scale changes in two-steps. First compute the Fourier transform of the patch, this transfers the information on translation in the log-polar patch into the phase of the transform. Second, compute the magnitude of each sample in the Fourier transformed patch, thereby removing the phase and thus the information on translation from the patch. We now have a local feature that is invariant to rotation and scale. This corresponds to the Fourier-Mellin transform used in [1], however the differences in the approaches are that we use local features and also we will not use phase information to recover the scaling and rotation.

## 2.2 Training

During training, the system does the following until all training images have been processed:

1. Read in next training image together with pose ground truth.
2. Detect Harris points and sample around each found Harris point with a log-polar sampling grid.
3. Store the Fourier transform of each feature patch together with information on the pose and the position it was found at in the database.

It should be noted that we do not perform the second step to make the feature invariant as detailed in Section 2.1 at this stage.

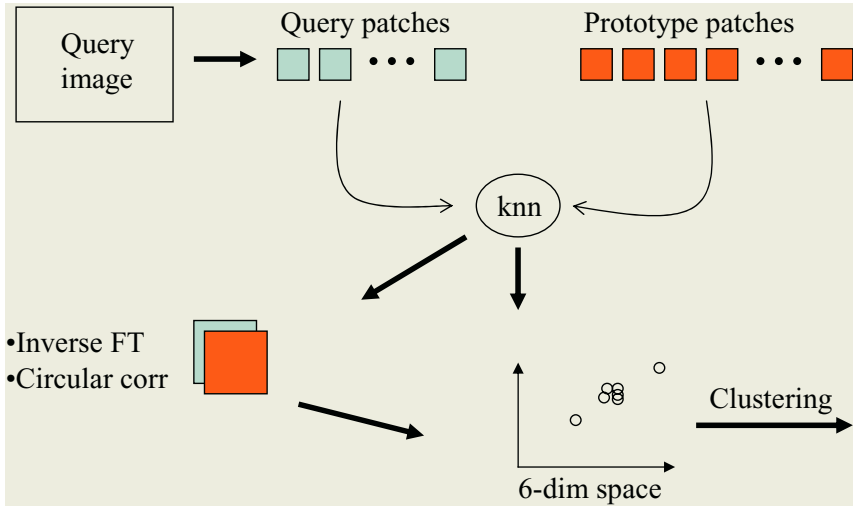
One advantage of this kind of method is that since we use no kind of optimization, as we would if we used a neural network, new views can be added at a later time. Storing data in such a database can be seen as a crude way of doing learning, there is however evidence that the human vision system works as if it used database look-up functions when recognizing objects [3].

### 2.3 Matching

When the system is running in query mode, i.e. it has already been trained and we want to use it to estimate the state of an object, we need to perform matching to see what votes will be cast. The matching procedure can be visualized by Figure 3. The second step detailed in Section 2.1, which is performed to make the features invariant to scale and rotation, is applied. Correlation is then used to compute the  $k$  nearest neighboring matches between the query and prototype features. The  $k$  nearest prototypes to each query feature are selected to cast a vote. The vote on pose angle as well as position is given by their position in the database. To compute the votes for scale and rotation angle we apply the inverse Fourier transform on the selected query and prototype patches to again get the log-polar transform. A modified circular correlation between each query feature and its  $k$  nearest neighboring prototype features yields a response where we can find the votes on scale and rotation by locating the peak in that output.

### 2.4 Clustering

The votes for  $\phi$ ,  $\theta$ , rotation in the image plane  $\alpha$ , position and scale are inserted into a 6-dimensional space. We need to find peaks in this space and estimate a mean of such a peak, or cluster. For this, mean-shift clustering [4],[2] is used. The algorithm finds one or many clusters and outputs a confidence value for each cluster that depends on how many votes there are in that specific cluster and how spread out the votes are. This means that the method can be used to search for several objects of the same kind as they will form different clusters since it is not physically possible for two objects to have the same exact state. Furthermore it should be realized that the method takes longer time to compute the more features are detected, for instance in the background, and that the more random and erroneous features there are, the higher the probability of erroneous clusters forming by chance will be.



**Fig. 3.** Overview of the query mode. The resulting output is an estimated pose, position, rotation, and scale of the object. KNN refers to the  $k$  nearest neighbor method

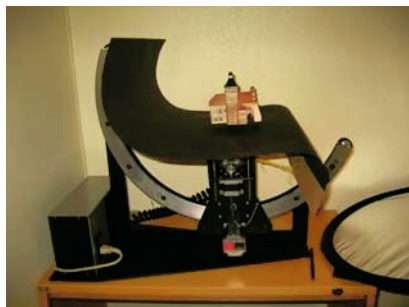
### 3 Experiments and Results

As we want to try and estimate the pose of an object we used the turntable seen in Figure 4 to sample a set of images. The turntable can be controlled very precisely and can rotate about two axes. The  $\phi$ -axis does however tend to align with the optical axis of the camera at high  $\theta$  angles. This alignment means that rotations in  $\phi$  can be mistaken for rotations in  $\alpha$ , i.e rotations in the image plane. We are using a feature that is supposed to be invariant to rotations in the image plane and this is the reason why the  $\theta$  angle in the data sets does not go as high as it could. A subset of the sampled images of a toy car can be seen in Figure 5, where the  $\theta$  angle is on the vertical axis and the  $\phi$  angle is on the horizontal axis. From this set of sampled images we define the following data sets

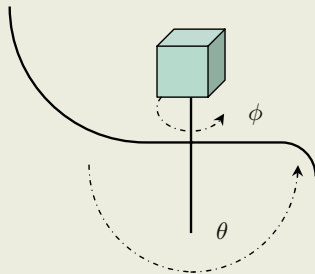
		$\phi$	$\theta$
Dataset 1	Training	$0^\circ, 10^\circ, \dots, 180^\circ$	$0^\circ, 10^\circ, \dots, 40^\circ$
	Evaluation	$5^\circ, 15^\circ, \dots, 175^\circ$	$5^\circ, 15^\circ, \dots, 35^\circ$
Dataset 2	Training	$0^\circ, 20^\circ, \dots, 180^\circ$	$0^\circ, 20^\circ, 40^\circ$
	Evaluation	$10^\circ, 30^\circ, \dots, 170^\circ$	$10^\circ, 30^\circ$

and evaluation on these two data sets yielded the following mean absolute errors

		Single angle	Double angle	Patch duplets [8]
Dataset 1	$\phi$	$0.53^\circ$	$0.48^\circ$	$1.25^\circ$
	$\theta$	$0.85^\circ$	$0.80^\circ$	$1.06^\circ$
Dataset 2	$\phi$	$2.42^\circ$	$1.84^\circ$	$4.21^\circ$
	$\theta$	$2.26^\circ$	$1.63^\circ$	$2.66^\circ$



(a) Turntable



(b) Possible rotation angles

**Fig. 4.** Turntable used to sample images**Fig. 5.** Subset of the training images of the toy car

From the above table we can see that for this particular set of images this method is comparable to the patch duplets [8], which is in the right-most column. We also see that the double angle representation seem to be better suited for this task than the single angle representation.

To find out how this method behaves on images with structured background we made some other experiments. Since we do not have ground truth in the following experiments we choose to overlay the query image with an edge-filtered version of the closest training view. Since we only have views with a  $5^\circ$  interval we can have errors up to that level even though the estimates might be more precise than that. One experiment can be seen in Figure 6, where the scale was found to be 1.1 and the overlay was scaled accordingly. An other such experiment can be seen in Figure 7, where scale was detected as 1.05. The experiment seen in Figure 8 also shows that the method is robust to some occlusion.

## 4 Conclusions

It is obvious from images presented in Section 3 that the proposed local feature together with the described matching and clustering works for real-world objects.

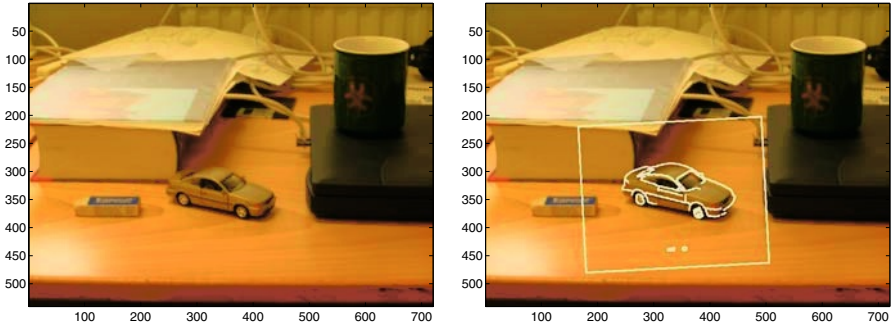


Fig. 6. Toy car on table and closest view overlaid

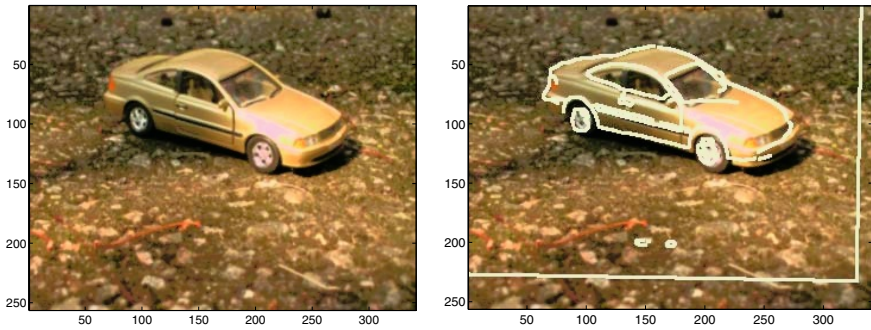


Fig. 7. Toy car on asphalt and closest view overlaid

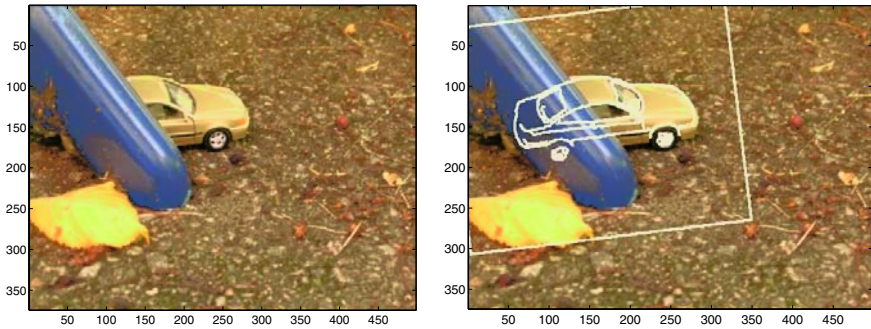


Fig. 8. Toy car occluded and closest view overlaid

It is also seen that in some cases the method is more precise than the method it is compared to. Since the properties of the feature allows the method to use only single patches, in contrast to for example [6] or [8], it has the chance to be more stable to occlusion than non-single-patch features. The single-patch property might also make it possible for the method to generalize to similar objects, which can be a good thing in some cases.

## References

1. Q. Chen, M. Defrise, and F. Deconinck. Symmetric phase-only matched filtering of fourier-mellin transforms for image registration and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-16(12), 1994.
2. Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, August 1995.
3. S. Edelman and H. Bulthoff. Modeling human visual object recognition. In *Proc. International Joint Conference on Neural Networks*, volume 4, pages 37–42, September 1992.
4. Keinosuke Fukunaga and Larry D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.
5. G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.
6. G. H. Granlund and A. Moe. Unrestricted recognition of 3-D objects for robotics using multi-level triplet invariants. *Artificial Intelligence Magazine*, 25(2):51–67, 2004.
7. C. G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, September 1988.
8. Björn Johansson and Anders Moe. Patch-duplets for object recognition and pose estimation. Technical Report LiTH-ISY-R-2553, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, November 2003.
9. Tony Lindeberg. *Scale-space Theory in Computer Vision*. Kluwer Academic Publishers, 1994. ISBN 0792394186.
10. D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV'99*, 1999.
11. E. Rivlin and H. Rotstein. Control of a camera for active vision: Foveal vision, smooth tracking and saccade. *IJCV*, 39(2):81–96, September 2000.
12. C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.
13. C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Int. Journal of Computer Vision*, 37(2):151–172, 2000.



# Dealing with Multiple Motions in Optical Flow Estimation

Jesús Chamorro-Martínez, Javier Martínez-Baena,  
Elena Galán-Perales, and Beén Prados-Suárez\*

Department of Computer Science and Artificial Intelligence  
University of Granada, Spain  
{jesus, jbaena, elena, belenps}@decsai.ugr.es

**Abstract.** In this paper, a new approach to optical flow estimation in presence of multiple motions is presented. Firstly, motions are segmented on the basis of a frequency-based approach that groups spatio-temporal filter responses with continuity in its motion (each group will define a *motion pattern*). Then, the gradient constraint is applied to the output of each filter so that multiple estimations of the velocity at the same location may be obtained. For each “motion pattern”, the velocities at a given point are then combined using a probabilistic approach. The use of “motion patterns” allows multiple velocities to be represented, while the combination of estimations from different filters helps reduce the aperture problem.

**Keywords:** Optical flow, multiple motions, spatio-temporal models

## 1 Introduction

Optical flow estimation, viewed as an approximation to image motion, is a very useful task in video processing [1]. In this framework, an open problem is how to deal with the presence of multiple motions at the same location [2]. With the presence of occlusions and transparencies, more than one velocity may be presented at the same point (for example, let us consider a sheet of glass crossing over an opaque object). In such cases, the techniques which do not consider the presence of multiple motions will generate erroneous estimations which will combine into a single vector the different velocities present at one point. These problems are currently being addressed by the research community with models such as those based on the use of mixed velocity distributions (usually two) at each point [3], the models based on line processes [4], the parametric models [5] or the frequency-based techniques (which use spatio-temporal filters to separate the motions [6, 7]). Nevertheless, although they do consider the presence of occlusions and transparencies in their calculations, the majority of these techniques do not generate a representation as an output which allows more than one velocity per point.

---

\* This work has been supported by the MCYT (Spain) under grant TIC2003-01504.

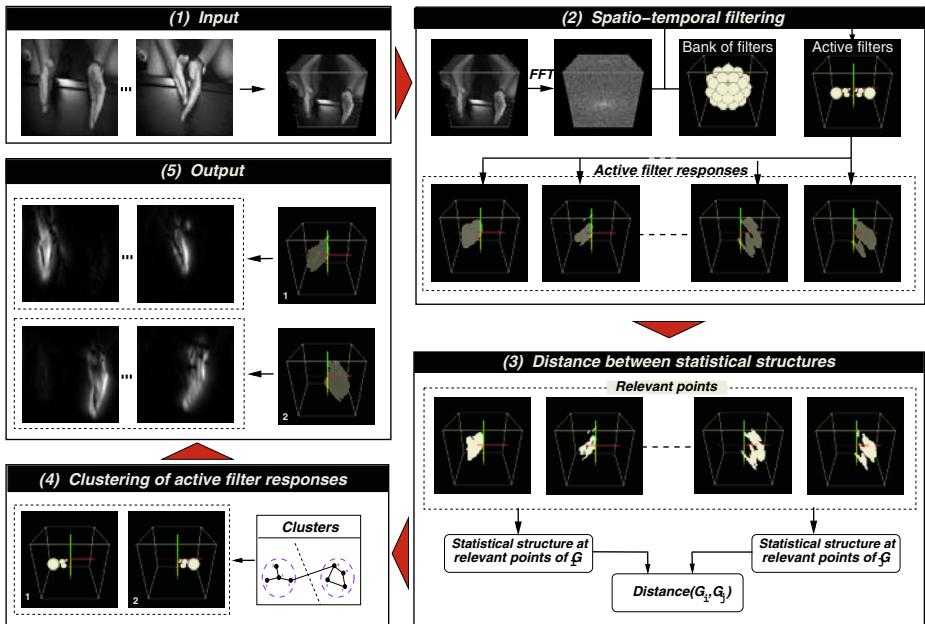


Fig. 1. A general diagram describing the motion segmentation model.

In order to confront this problem, in this paper we develop a methodology for optical flow estimation that is able to represent multiple velocities at the same point. To detect points with multiple motions, the model introduced in [8] is used. This model is a frequency-based approach that groups spatio-temporal filter responses with continuity in its motion (each group will define a *motion pattern*). Given a motion pattern (a group of filters), the proposed technique apply the gradient constraints to the output of each filter in order to obtain multiple estimates of the velocity at the same location. Then velocities at each point are combined using probability rules.

The rest of the paper is organized as follows. Section 2 introduces the spatio-temporal filtering approach to motion segmentation and Section 3 shows its application to optical flow estimation with of multiple motions. Results with real and synthetic sequences are shown in Section 4 and, finally, the main conclusions are summarized in Section 5.

## 2 Motion Patterns

To detect multiple motions at the same location, the frequency-domain approach introduced in [8] is used. This methodology is based on three main stages: a spatio-temporal filtering, the computation of the distance between filter responses, and a clustering process. A diagram illustrating the analysis on a sequence corresponding to a handclap is shown in Figure 1 (in this example, the objective is to separate the two hand motions).

In the first stage, the original sequence is represented as a spatio-temporal volume, where a moving object corresponds to a three-dimensional pattern. Its Fourier transform is then calculated in order to perform the analysis in the frequential domain. Given a bank of spatio-temporal logGabor filters, a subset of these is selected so that significant spectral information may be extracted. These selected filters are applied over the original spatio-temporal image so that a set of active responses may be obtained (only one subset of filters is used).

In the second stage, the distances between active filters are obtained. These distances are computed over relevant points which are calculated as local energy peaks on the filter response.

In the third stage, a clustering over the set of active filters is performed to highlight response invariance. Each cluster obtained in this step defines a motion pattern. In the output box of Figure 1, two collections of filters corresponding to the two hand motions are shown. For more details about these three stages, see [8].

### 3 Optical Flow Estimation

In this section, the frequency-based model described in Section 2 will be used to optical flow estimation in presence of multiple motions

#### 3.1 Differential Formulation

Within the gradient-based approaches, based on the well known differential brightness constancy constraint equation, a probabilistic framework to optical flow estimation was proposed by Simoncelli et al. [9]. In this approach, two independent additive Gaussian noise terms  $\mathbf{n}_1$  and  $n_2$  are introduced in the constancy constraint equation [9], and the velocity at a given point is defined as a Gaussian random variable with mean and covariance:

$$\mu_{\mathbf{v}} = -\Delta_{\mathbf{v}} \cdot \sum_r \frac{w_r \mathbf{d}_r}{\kappa_1 \|\mathbf{f}_{\mathbf{e}}(x_r, y_r, t)\|^2 + \kappa_2} \quad (1)$$

$$\Delta_{\mathbf{v}} = \left[ \sum_r \frac{w_r \mathbf{M}_r}{\kappa_1 \|\mathbf{f}_{\mathbf{e}}(x_r, y_r, t)\|^2 + \kappa_2} + \Delta_p^{-1} \right]^{-1} \quad (2)$$

with  $\mathbf{f}_{\mathbf{e}} = (f_x, f_y)$  and  $f_t$  being, respectively, the spatial and temporal derivatives, where  $w_r$  is a weighting function that gives more influence to elements at the center of the neighborhood, with the points in the neighborhood indexed by  $r$ ,  $\Delta_p$  the covariance of the prior distribution  $P(\mathbf{v})$ ,  $\mathbf{M}_r$  and  $\mathbf{d}_r$  defined as

$$\mathbf{M}_r = \begin{bmatrix} f_x^2(r) & f_x(r)f_y(r) \\ f_y(r)f_x(r) & f_y^2(r) \end{bmatrix} \quad \mathbf{b}_r = \begin{bmatrix} f_x(r)f_t(r) \\ f_y(r)f_t(r) \end{bmatrix} \quad (3)$$

and  $\kappa_1$  and  $\kappa_2$  two parameters associated to  $\mathbf{n}_1$  and  $n_2$  respectively (see [9] for more details)

### 3.2 Estimation for a Spatio-temporal Filter Response

In order to estimate the velocity  $\mathbf{v}_i$  at a given point  $(x, y, t)$  of the  $i$ -th filter  $\phi_i$ , the probabilistic approach described in Section 3.1 is used. Using the odd response of the filter  $\phi_i$ , the velocity  $\mathbf{v}_i$  is therefore defined on the basis of a Gaussian random variable  $\mathbf{v}_i$  with mean  $\mu_{\mathbf{v}_i}$  and covariance  $\Delta_{\mathbf{v}_i}$ :

$$\mathbf{v}_i \sim N(\mu_{\mathbf{v}_i}, \Delta_{\mathbf{v}_i}) \quad i = 1, \dots, N \quad (4)$$

where  $\mu_{\mathbf{v}_i}$  and  $\Delta_{\mathbf{v}_i}$  are calculated using Equations (1) and (2). Therefore, given a point  $(x, y, t)$ , we shall have a vector of estimations  $[\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N]$ , with  $N$  being the number of active filters

**Confidence Measure** It is well known that the covariance matrix  $\Delta_{\mathbf{v}_i}$  can be used to define a confidence measure of the estimation  $\mathbf{v}_i$  [9]. In this paper, we shall use the smallest eigenvalue of  $\Delta_{\mathbf{v}_i}^{-1}$  as the confidence measure of  $\mathbf{v}_i$  [10] and this will be denoted  $\lambda_{\mathbf{v}_i}$ :

$$\lambda_{\mathbf{v}_i} = \min \{ \lambda_1^i, \lambda_2^i \} \quad (5)$$

where  $\lambda_1^i$  and  $\lambda_2^i$  are the two eigenvalues of  $\Delta_{\mathbf{v}_i}^{-1}$  (for the sake of simplicity, we have omitted the spatio-temporal parameters  $(x, y, t)$  in the notation  $\lambda_{\mathbf{v}_i}(x, y, t)$ ).

Therefore, an estimation  $\mathbf{v}_i$  at a given point  $(x, y, t)$  of the  $i$ -th filter  $\phi_i$  will be accepted if  $\lambda_{\mathbf{v}_i} \geq \text{Threshold}_{\phi_i}$ , where  $\text{Threshold}_{\phi_i}$  is a confidence threshold associated to the filter  $\phi_i$ . Under the assumption that every relevant point of the filter will generate a reliable estimation, the following approximation is proposed to calculate  $\text{Threshold}_{\phi_i}$ :

$$\text{Threshold}_{\phi_i} = \min \{ \lambda_{\mathbf{v}_i}(x, y, t) / (x, y, t) \in P(\phi_i) \} \quad (6)$$

where  $P(\phi_i)$  represents the set of relevant points of the filter  $\phi_i$ . In this way, we accept as reliable any estimation which is the same as or better than the worst estimation obtained for the set of relevant points.

### 3.3 Estimation for a Motion Pattern

This section shall describe the methodology for integrating the estimations corresponding to the set of filters which comprise a motion pattern. Let  $S_k$  be the  $k$ -th motion pattern detected in the sequence, and let  $\{\phi_i^k\}^{i=1, \dots, L_k}$  be the set of  $L_k$  grouped filters in  $S_k$ . Let  $\Omega_k$  be the set of estimations  $\mathbf{v}_i \sim N(\mu_{\mathbf{v}_i}, \Delta_{\mathbf{v}_i})$  obtained from  $\{\phi_i^k\}^{i=1, \dots, L_k}$  which are above the confidence threshold. The integration will be performed on the basis of a linear combination

$$\hat{\mathbf{v}}_k = \sum_{\mathbf{v}_i \in \Omega_k} \alpha_i \mathbf{v}_i \quad (7)$$

with  $\hat{\mathbf{v}}_k$  representing the velocity at point  $(x, y, t)$  of the motion pattern  $P_k$ , and  $\alpha_i$  given by the equation

$$\alpha_i = \frac{\|\mu_{\mathbf{v}_i}\| \lambda_{\mathbf{v}_i}}{\sum_{\mathbf{v}_j \in \Omega_k} \|\mu_{\mathbf{v}_j}\| \lambda_{\mathbf{v}_j}} \quad (8)$$

In this equation, the norm  $\|\mu_{\mathbf{v}_i}\|$  measures the “amount of motion” detected at this point by the filter  $\phi_i$ , while  $\lambda_{\mathbf{v}_i}$  measures the reliability of the estimation  $\mathbf{v}_i$  (Equation (5)). The denominator in (8) guarantees that  $\sum_{\Omega_k} \alpha_i = 1$ .

If we assume that  $\mathbf{v}_i$  are independent variables,  $\widehat{\mathbf{v}}_k$  will be a random variable with a Gaussian distribution with mean  $\mu_{\widehat{\mathbf{v}}_k} = \sum_{\Omega_k} \alpha_i \mu_{\mathbf{v}_i}$  and covariance  $\Delta_{\widehat{\mathbf{v}}_k} = \sum_{\Omega_k} \alpha_i^2 \Delta_{\mathbf{v}_i}$ .

### 3.4 Representation of Multiple Velocities

The motion patterns allow the relevant motions presented in a given sequence to be separated; therefore, in the optical flow estimation problem, they can be used to decide whether there are multiple velocities at the same location or not. Based on this idea, our scheme will obtain the velocities at a given point  $(x, y, t)$  directly from the estimations calculated for each motion pattern as:

$$\bar{\mathbf{v}} = \{\widehat{\mathbf{v}}_k\}_{k=1\dots K} \quad (9)$$

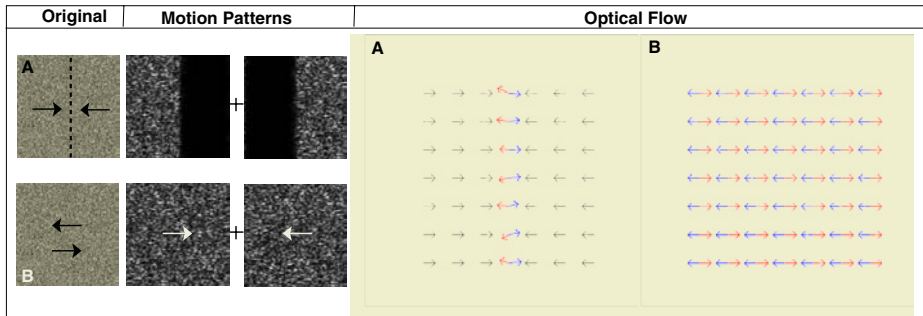
where  $K$  is the number of motion patterns detected in the sequence, and  $\widehat{\mathbf{v}}_k$  is the optical flow estimation at point  $(x, y, t)$  of the  $k$ -th motion pattern  $S_k$ . It should be noted that due to the use of confidence measures, we will not always have  $K$  estimations at each point.

## 4 Results

### 4.1 Synthetic Sequences

Figure 2 shows two synthetic sequences which have been generated with Gaussian noise of mean 1 and variance 0. The first example (Figure 2(A)) shows a sequence where a background pattern with velocity  $(-1,0)$  frames/image is occluded by a foreground pattern with velocity  $(1,0)$ . The second example (Figure 2(B)) shows two motions with transparency: an opaque background pattern with velocity  $(1,0)$  and a transparent foreground pattern with velocity  $(-1,0)$ . In both cases, the figure shows the central frame of the sequence, the motion patterns detected by the model (two in each case), and the optical flow estimated with our technique. In this example, we have used the values  $\kappa_1 = 0$ ,  $\kappa_2 = 1$  and  $\kappa_p = 1e - 5$  (with  $\Delta_p^{-1} = \kappa_p I$  [9]) in Equations (1) and (2) as it is proposed in [9], the spatial and temporal partial derivatives have been calculated using the kernel  $\frac{1}{12}(-1, 8, 0, -8, 1)$ , the gradient constraints have been applied in a local neighborhood of size  $5 \times 5$ , and the weight vector has been fixed to  $(0.0625, 0.25, 0.375, 0.25, 0.0625)$  [10].

We should point out that in the first example, our technique obtains two velocities at the occlusion points; in a similar way, in the second example, our methodology is able to estimate two velocities for each point of the frame. Since we have access to the true motion field of the synthetic sequences, we can measure the angular error [10]. Table 1 shows a comparison between our methodology and other classic techniques such as those studied by Barron et al. [10].



**Fig. 2.** Results with synthetic sequences.

**Table 1.** Mean error obtained with several techniques applied to the sequences in Figure 2. MV: Multiple velocities. SV: Single velocity. Density is 100%.

		A (occlusion)	B (transparency)
Proposed technique	MV	0.84°	0.44°
Nestares	MV	3.93°	7.76°
Lucas&Kanade	SV	4.79°	50.89°
Horn&Schunk	SV	2.66°	52.77°
Nagel	SV	8.59°	45.81°
Anandan	SV	10.47°	47.78°
Singh	SV	2.97°	45.27°
Uras	SV	3.96°	57.86°
Simoncelli	SV	5.97°	49.38°

## 4.2 Real Sequences

Figure 3 shows some examples with real sequences. In this case, we have used the values  $\kappa_1 = 0$ ,  $\kappa_2 = 1$  and  $\kappa_p = 0.5$  (as it is proposed in [9]) with the same partial derivatives and weight parameters used in the synthetic case. For each example, the figure shows the first and last frame of the original sequence, the motion patterns detected in each case, the optical flow estimated with our technique and the optical flow estimated employing the Simoncelli's technique [9] as described in section 3.1 (which uses a similar approach, but without a multiple velocity representation). As we do not have the true motion field for real image sequences, we can only show the computed flow field.

The first example (Figure 3(A)) shows a case of occlusion where a hand is crossing over another one. The second case shows an example of transparency where a bar is occluded by a transparent object (Figure 3(B)). Finally, Figure 3(C) shows an example with an articulated object with two components rotating and approaching independently. In all the cases, our methodology extracts two motion patterns and estimates two velocities in the occlusion points.

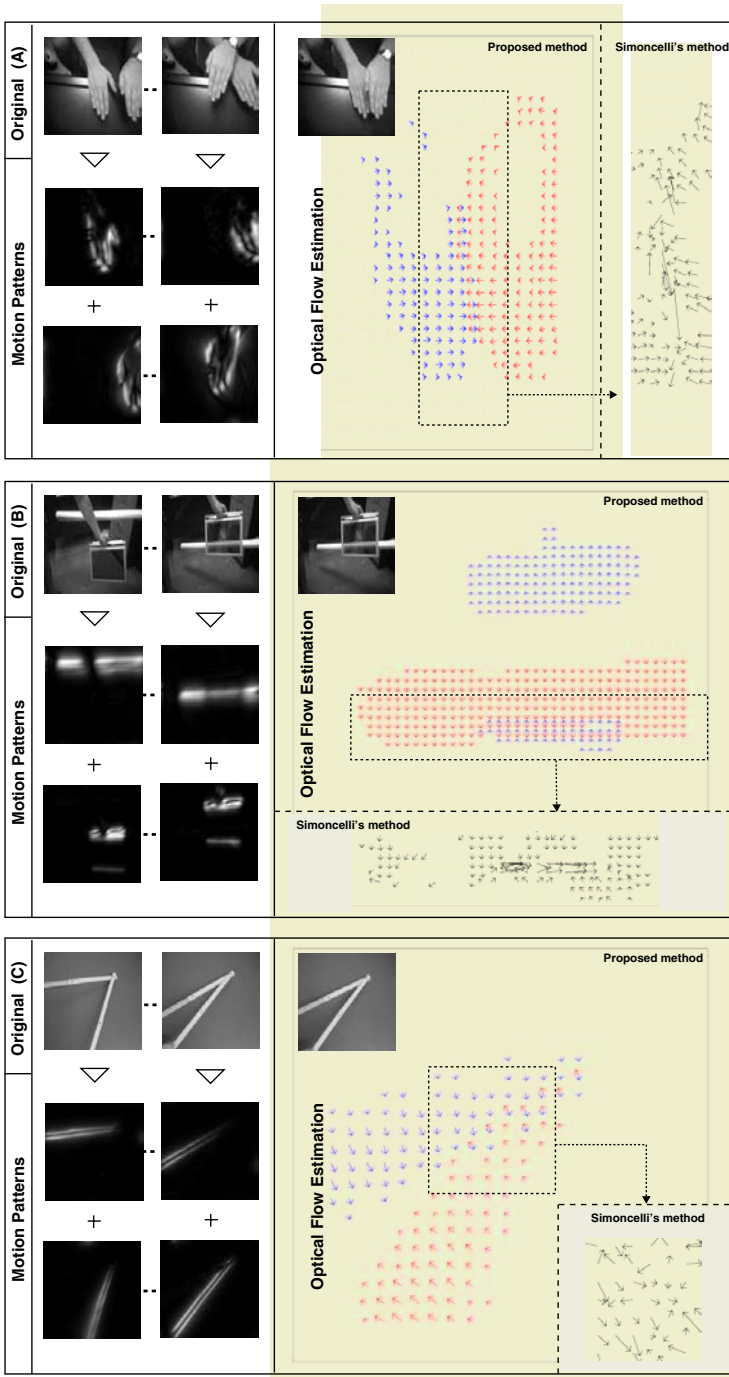


Fig. 3. Results with real sequences.

## 5 Conclusions

In this paper, a new methodology for optical flow estimation has been presented. The proposed technique is able to represent multiple velocities on the basis of a new frequency-domain approach capable to detect “motion patterns” (that is, a clustering of spatio-temporal filter responses with continuity in its motion). A methodology to obtain the optical flow corresponding to a spatio-temporal filter response has been proposed, using confidence measures to ensure only reliable estimations. A probabilistic combination of velocities corresponding to the set of filters clustering in a given motion pattern has been proposed. One of the main features of the proposal is the possibility of representing more than one velocity at a point. This is extremely important in situations with occlusions or transparencies.

## References

1. B.K.P. Horn and B.G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
2. M. Pingault, E. Bruno, and D. Pellerin, “A robust multiscale b-spline function decomposition for estimating motion transparency,” *IEEE Transactions on Image Processing*, vol. 12, pp. 1416–1426, 2003.
3. B.G. Schunck, “Image flow segmentation and estimation by constraint line clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 10, pp. 1010–1027, 1989.
4. H. Nagel and W. Enkelmann, “An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 565–593, 1986.
5. M.J. Black and P. Anandan, “The robust estimation of multiple motion: parametric and piecewisewise smooth flow fields,” *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.
6. D.J. Heeger, “Model for the extraction of image flow,” *Journal of the Optical Society of America A*, vol. 4, no. 8, pp. 1455–1571, 1987.
7. O. Nestares and R. Navarro, “Probabilistic estimation of optical flow in multiple band-pass directional channels,” *Image and Vision Computing*, vol. 19, no. 6, pp. 339–351, 2001.
8. J. Chamorro-Martinez, J. Fdez-Valdivia, and J. Martinez-Baena, “A spatio-temporal filtering approach to motion segmentation,” in *Pattern Recognition and Image Analysis*, F.J. Perales, A.J.C. Campilho, N. Perez de la Blanca, and A. Sanfeliu, Eds., Lecture Notes in Computer Science LNCS 2652, pp. 193–203. Springer, June 2003.
9. E.P. Simoncelli, E.H. Adelson, and D.J. Heeger, “Probability distributions of optical flow,” *IEEE Proceedings of CVPR’91*, pp. 310–315, 1991.
10. J.L. Barron, D.J. Fleet, and S. Beauchemin, “Performance of optical flow techniques,” *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.



# Conversion into Three-Dimensional Implicit Surface Representation from *Topological Active Volumes* Based Segmentation

José Rouco<sup>1</sup>, Noelia Barreira<sup>1</sup>, Manuel G. Penedo<sup>1</sup>, and Xosé M. Pardo<sup>2</sup>

<sup>1</sup> Dpto. Computación. Fac. Informática, Universidade da Coruña  
15071 A Coruña, Spain  
insjrm00@ucv.udc.es, {noelia,cipenedo}@dc.fi.udc.es

<sup>2</sup> Dpto. Electrónica e Computación, Universidade de Santiago de Compostela  
15782 Santiago de Compostela, Spain  
pardo@dec.usc.es

**Abstract.** In the last few years, the advances in three-dimensional medical image processing have made possible operations like planning or simulation over real data. Different representations of structures or models have been proposed, being the implicit surfaces one of the most flexible models for processing. This paper introduces a new method for computing the implicit surfaces from the explicit representations of the objects segmented in three-dimensional images. This proposal is based on the approximation of the surfaces using distance functions and *natural neighbor interpolation*. The system has been tested over *CT* images of tibia and femur where the explicit representation has been extracted through a *TAV* model [1]. The results obtained show the suitability of the method for a correct representation of the target objects.

## 1 Introduction

Three-dimensional image data from *magnetic resonance imaging (MRI)*, *computed tomography (CT)* and other scanning techniques allow scientist to interact with anatomical structures directly mapped from patients. In the last few years, medical imaging has expanded its use to new applications like surgical planning an simulation, where a good representation of the organs is necessary. In such applications, the implicit object representations are adequate due to their suitability for collision detection and physically based animation. These two features form the basis for intuitive and realistic interaction with solid objects. However, most of these applications use segmentation processes for the extraction of the target objects, and most of these processes produce explicit representations of the surfaces (like polygonal meshes or unorganised points) that must be converted to implicit representations.

There are several techniques for conversion of explicit to implicit representation. The method based on *scan conversion*, sample the surface into a binary volume and then apply a *distance transformation algorithm* [9]. Breen et al. [4]

use a similar idea, first sampling the distance to surface into a *narrow band* near it and then propagating this information using *fast marching method* [7]. Some approaches use implicit function fitting: *radial basis functions* [10], *moving least squares* [8] and *level set methods* [7] are also used for surface interpolation and fitting. Finally, geometric approaches have been proposed, some of them based on the identification of the vertex, edges and facets closest to regions in space [4] while others use *Voronoi diagram* for *natural neighbor interpolation* of distance functions associated to points on the surface [3, 6].

This paper introduces a framework where the target objects of three-dimensional scenes are extracted using the *Topological Active Volume (TAV)* model [1]. The result of this process is: a set of points on the surface of the objects, another set of points inside the object and the topological relations between them. From this information, we approximate the implicit functions representing the objects through *natural neighbor interpolation* of distance functions [3]. The paper is organised as follows. Section 2 introduces the *TAV* model. Section 3 describes method for the reconstruction of implicit surfaces and how the *TAV* model is adapted to it. Section 4 shows our preliminary results. And section 5 exposes the conclusions from our work.

## 2 Topological Active Volumes (TAV)

The *Topological Active Volumes (TAV)* model is an active contour model focused on extraction and modelisation of volumetric objects in three-dimensional scenes [1]. A *Topological Active Volume* is a three-dimensional structure composed of interrelated nodes where the basic repeated structure is a cube. There are two kinds of nodes: the external nodes, that fit the surface of the object, and the internal nodes, that model its internal topology. The state of the model is governed by an energy function defined as follows:

$$E(v) = \int_0^1 \int_0^1 \int_0^1 E_{int}(v(r, s, t)) + E_{ext}(v(r, s, t)) dr ds dt \quad (1)$$

where  $E_{int}$  and  $E_{ext}$  are the internal and the external energy of the *TAV*, respectively. The internal energy controls the shape and the structure of the net. Its calculation depends on first and second order derivatives that control contraction and bending, respectively. It is defined by the following equation:

$$\begin{aligned} E_{int}(v(r, s, t)) = & \alpha(|v_r(r, s, t)|^2 + |v_s(r, s, t)|^2 + |v_t(r, s, t)|^2) + \\ & \beta(|v_{rr}(r, s, t)|^2 + |v_{ss}(r, s, t)|^2 + |v_{tt}(r, s, t)|^2) + \\ & 2\gamma(|v_{rs}(r, s, t)|^2 + |v_{rt}(r, s, t)|^2 + |v_{st}(r, s, t)|^2) \end{aligned} \quad (2)$$

where subscripts represents partial derivatives and  $\alpha$ ,  $\beta$  and  $\gamma$  are coefficients controlling the first and second order smoothness of the net.

$E_{ext}$  represents the features of the scene that guide the adjustment process and is different for external and internal nodes. It is defined as:

$$E_{ext}(v(r, s, t)) = \omega f[I(v(r, s, t))] + \frac{\rho}{\aleph(r, s, t)} \sum_{n \in \aleph(r, s, t)} \frac{1}{\|v(r, s, t) - v(n)\|} f[I(v(n))] \quad (3)$$

where  $\omega$  and  $\rho$  are weights,  $I(v(r, s, t))$  is the intensity value of the original image in the position  $v(r, s, t)$ ,  $\aleph(r, s, t)$  is the neighbourhood of the node  $(r, s, t)$  and  $f$  is a function of the image intensity, which is different for both types of nodes. For example, if the objects to detect are light and the background is dark, function  $f$  is defined as follows in order to minimise the energy value of external and internal nodes when they are on the surface or inside the objects, respectively:

$$f[I(v(r, s, t))] = \begin{cases} h[\overline{I_{max} - I_N(v(r, s, t))}] & \text{for internal nodes} \\ h[\overline{I_N(v(r, s, t))} + \xi(G_{max} - G(v(r, s, t)))] & \text{for external nodes} \\ + DG(v(r, s, t)) & \end{cases} \quad (4)$$

$\xi$  is a weighting term;  $I_{max}$  and  $G_{max}$  are the maximum intensity values of image  $I$  and the gradient image  $G$ , respectively;  $I(v(r, s, t))$  and  $G(v(r, s, t))$  are the intensity values of the original and gradient image in the position  $v(r, s, t)$ ;  $\overline{I_N(v(r, s, t))}$  is the mean intensity in a  $N \times N \times N$  cube and  $h$  is an appropriate scaling function;  $DG(v(r, s, t))$  is the distance from the position  $v(r, s, t)$  to the nearest position in the gradient image that points out an edge.

The *TAV* model is automatic, so the initialisation does not need any human interaction as other deformable models. As a broad outline, the adjustment process consists of the minimisation of the energy of the mesh and, after that, the breaking of connections between external nodes badly placed, this is, the external nodes that are not on the surfaces of the objects. The breaking of connections allows a perfect adjustment to the surfaces and the detection of holes and several objects in the three-dimensional scene [1].

### 3 Implicit Surface Reconstruction

#### 3.1 Approximating Distance Functions

The most common approach for three-dimensional object surface representation is the explicit (or parametric) model. In this kind of representation we can easily identify point coordinates on the object's surface by varying its parameters. Opposing to this, in an implicit representation, points on the surface are those that satisfy an equation like  $F(x, y, z) = 0$ , where  $F(x, y, z)$  is the so called *implicit function*. Thus, the surface  $F(x, y, z) = 0$  divides the space in two areas, one where  $F(x, y, z) < 0$  and the other where  $F(x, y, z) > 0$ . This is often used for distinguish between the inside and the outside of the object using a function that takes negative values inside and positive outside (or vice versa).

When we try to convert an explicit representation, like a *TAV* into an implicit one, we first have to choose the *implicit function* to represent the object. In this paper we use the signed distance function to the surface  $D(p)$  as implicit function. This function is defined as the shortest distance from point  $p = \{x, y, z\}$  to any

point on the surface.  $D(p)$  is positive if  $p$  lies outside the object and negative if  $p$  lies inside it.

The *implicit function* we propose is derived from the interpolation of signed distance functions associated to the points that are known to be on the object surface. These functions are approximations of the distance to the object surface around the points. Our approach uses *natural neighbor interpolation* of these distance functions, as it has proved its suitability for surface reconstruction ([3], [6]) and it is guaranteed to produce correct results when the sampling density increases enough.

*Natural neighbor interpolation* is a weighted average of the values at the neighbour data points using *natural coordinates* as the weighting measure. Let  $\mathcal{S}$  be a set of points  $s_i$  where the function to be interpolated is known (we know the local distance function from any point  $p$  to the surface  $d_{s_i}(p)$  at  $s_i$  nearness), and  $\mathcal{V}_S$  the *Voronoi diagram* of the data sites. The *natural neighbors* of any point  $p$  in  $\mathcal{S}$  are those that are neighbours of  $p$  in  $\mathcal{V}_{(\mathcal{S} \cup p)}$ . For each  $s_i$  *natural neighbor* of  $p$ , the *natural region*  $NR_p^{s_i}$  is defined as the region of space that  $s_i$  loses when  $p$  is inserted in  $\mathcal{V}_S$ . Denoting  $d_{s_i}(p)$  as the distance function associated to  $s_i$  with respect to point  $p$ , the interpolated distance  $D(p)$  is computed as:

$$D(p) = \frac{\sum_{s_i \in \mathcal{S}} w_p^\sigma(s_i) d_{s_i}(p)}{\sum_{s_i \in \mathcal{S}} w_p^\sigma(s_i)} \quad (5)$$

$$w_p(s_i) = \frac{\mathcal{L}(\mathcal{V}_S(s_i) \cap \mathcal{V}_{(\mathcal{S} \cup p)}(p))}{\mathcal{L}(\mathcal{V}_{(\mathcal{S} \cup p)}(p))} \quad (6)$$

where  $\sigma \geq 1$  is the parameter that controls the relation between weight magnitude and point importance,  $w_p(s_i)$  is the natural coordinate of the point  $p$  associated to  $s_i$  and  $\mathcal{L}(R)$  denotes the *Lebesgue* measure of the region  $R$  (area in two dimensions, volume in three-dimensional space) and  $d_{s_i}(p)$  is the signed distance to the tangent plane at  $s_i$ . Thus, normal information is needed for each point. The denominator is added for weight normalisation in order to preserve the distance function magnitude when  $\sigma > 1$ .

### 3.2 Extracting Features from the TAV Model

The results from the segmentation process must be adapted to use the implicit surface reconstruction method described above. As preciously mentioned, this method uses points on the surface of the object and normal vectors to them. The main idea is to get the external node positions of the TAV as the set of points on the object surface and make an estimation of the normal (direction and orientation) using the topological information.

Let  $T$  be a TAV model and  $\mathcal{B}$  its boundary,  $\mathcal{N}_T|_{\mathcal{B}}$  denotes the set of external nodes of  $T$ . For each external node  $n = \{r, s, t\} \in \mathcal{N}_T|_{\mathcal{B}}$ ,  $v(n)$  denotes the node position and  $\mathcal{F}_T|_{\mathcal{B}}(n)$  the external facets adjacent to it. The number of adjacent external facets can vary from three to twelve due to the TAVs ability to make

topology changes, so we use the normal vector to all these facets for estimating the normal vector to  $\mathcal{B}$  around  $n$  proceeding as follows.

For each external node  $n \in \mathcal{N}_T|_{\mathcal{B}}$ , we compute the direction of the normal vectors  $\bar{N}'_{nf}$  associated to each facet  $f \in \mathcal{F}_T|_{\mathcal{B}}(n)$  as well as the displacement vectors  $\bar{d}_{nf}$  as follows:

$$\bar{N}'_{nf} = \text{norm}(\bar{e}_{nf}^1 \times \bar{e}_{nf}^2) \quad (7)$$

$$\bar{d}_{nf} = \text{norm}(\bar{e}_{nf}^1 + \bar{e}_{nf}^2) \quad (8)$$

where  $\bar{e}_{nf}^1$  and  $\bar{e}_{nf}^2$  are unitary vectors in the direction of the two edges of  $f$  adjacent to  $n$ , ' $\times$ ' denotes *vector product* and  $\text{norm}(\bar{v})$  normalises the vector  $\bar{v}$ . In order to ensure that the normals point to the outside of the object, the position of the *centroid*  $c_f$  of the cube  $\mathcal{C}_f$  that  $f$  belongs to is used ( $\mathcal{C}_f$  is unique since  $f$  is external). Then, the oriented normal  $\bar{N}_{nf}$  to each facet  $f$  associated to each of its nodes  $n$  is:

$$\bar{N}_{nf} = \begin{cases} -\bar{N}'_{nf} & \text{if } ((c_f - v(n)) \cdot \bar{N}'_{nf}) < 0 \\ \bar{N}'_{nf} & \text{otherwise} \end{cases} \quad (9)$$

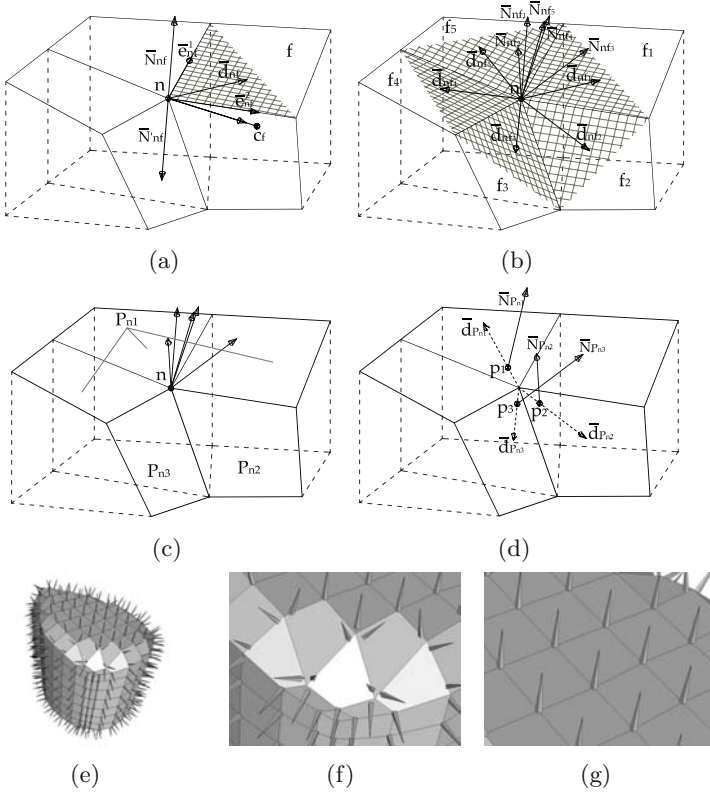
$$c_f = \frac{1}{8} \sum_{n_i \in \mathcal{C}_f} v(n_i) \quad (10)$$

where ' $\cdot$ ' denotes *dot product*. Note that vector  $(c_f - v(n))$  points inside the cube due to internal energy minimisation of the model. This minimisation should assure that the cubes are not degenerated. Hence, normal vector should have an angle of at least  $\frac{\pi}{2}$  radians with  $(c_f - v(n))$  and equation 9 gives the correct orientation. Figures 1(a) and 1(b) represent the vectors involved in this computation.

Normal orientation  $\bar{N}_{nf}$  to all facets  $f \in \mathcal{F}_T|_{\mathcal{B}}(n)$  associated to node  $n$  give us information about how surface  $\mathcal{B}$  varies around the node position  $v(n)$ . If these normals are similar, the surrounding surface can be approximated by a single plane. If not, see figure 1(c), such approximation differs from original surface and sub-sampling is needed near  $v(n)$ . With the aim of identifying the planes that have to be used for a good approximation of  $\mathcal{B}$  around  $v(n)$ , a divisive hierarchical cluster analysis algorithm (see [5]) over  $\mathcal{F}_T|_{\mathcal{B}}(n)$  is used. Neighbouring facets with similar orientation are good candidates for being grouped together, so we use the angle between normals as the dissimilarity measure and divide clusters if they have at least two normals with difference over a *threshold angle* ( $\theta$ ).

The result of this analysis is a partition of  $\mathcal{F}_T|_{\mathcal{B}}(n)$ , this is, a set of clusters  $\mathcal{P}_n = \{P_{n1}, P_{n2}, \dots, P_{nk}\}$  whose members are sets of similar facets that differ from the others. Thus, we approximate the surface around the node position using one plane for each of these clusters. This way, we choose the mean normal vector  $\bar{N}_{P_{ni}}$  of each cluster  $P_{ni} \in \mathcal{P}_n$  as its representing normal vector:

$$\bar{N}_{P_{ni}} = \text{norm}\left(\sum_{f \in P_{ni}} \bar{N}_{nf}\right) \quad (11)$$



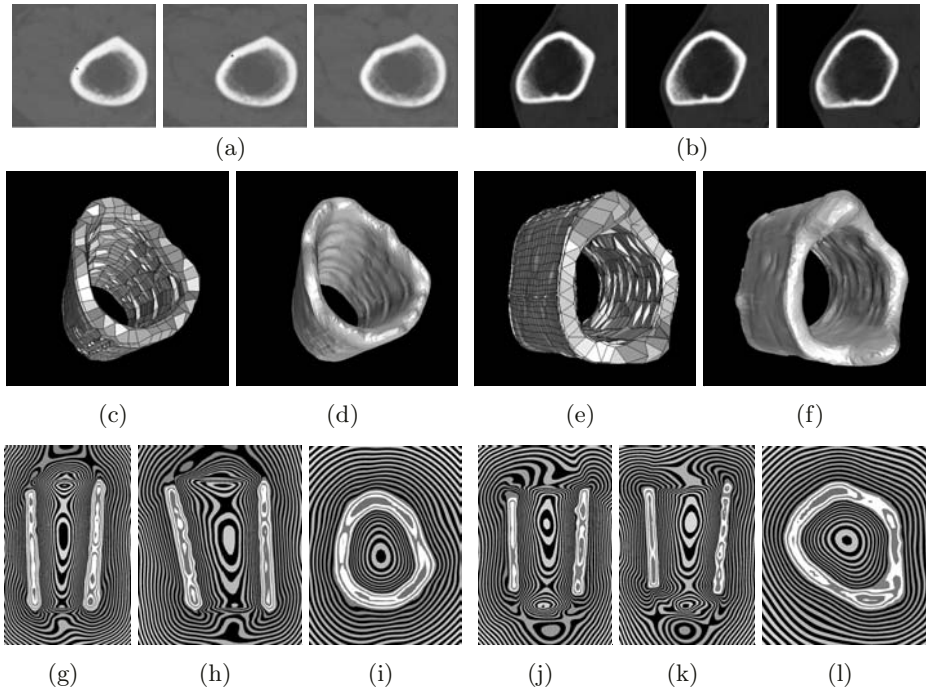
**Fig. 1.** Feature extraction from TAV. **(a)** Vectors involved in computation of  $\bar{N}_{nf}$  and  $\bar{d}_{nf}$ . **(b)**  $\bar{N}_{nf}$  and  $\bar{d}_{nf}$  for all facets in  $\mathcal{F}_T|_{\mathcal{B}}(n)$ . **(c)** Normals to neighbour facets and clusters. **(d)** Resulting points and normals. **(e)** Example of application. **(f)** Zoom on high variance area. **(g)** Zoom on low variance area.

being  $\bar{N}_{nf}$  the oriented normal to facet  $f$  associated to node  $n$ , and  $norm(\bar{v})$  a function that normalises  $\bar{v}$ . Analogously, for selecting the sample point  $p_{ni}$  we proceed as follows. If  $\mathcal{P}_n$  contains more than one cluster (figure 1(c)), a sample point  $p_{ni}$  is selected for each  $P_{ni}$  as:

$$p_{ni} = v(n) + \delta \bar{d}_{P_{ni}} \quad (12)$$

$$\bar{d}_{P_{ni}} = norm\left(\sum_{f \in P_{ni}} \bar{d}_{nf}\right) \quad (13)$$

where  $\delta$  is a displacement constant and  $\bar{d}_{P_{ni}}$  is the mean displacement vector of cluster  $P_{ni}$  (see figure 1(d)). If only one cluster is given, position  $v(n)$  is used as  $p_{ni}$ . Figures 1(e), 1(f) and 1(g) show an example of application of this method on a simple artificial object.



**Fig. 2.** Results. (a)(b) Femur and tibia CT slices. (c)(e) *TAV* results for femur and tibia. (d)(f) Zero level isosurfaces from femur and tibia resulting *implicit functions*. (g)(h)(i)(j)(k)(l)  $x, y$  and  $z$  sections from femur and tibia resulting *implicit functions*.

## 4 Results

We have used the proposed methodology for the segmentation of *CT* gray scale images that represent sections of the tibia and femur. Figures 2(a) and 2(b) show three of these slices from tibia and femur respectively.

The *CT* images (without filtering) were used to compute the external energy. The gradient images were obtained with a bi-dimensional *Sobel* filter. *TAV* parameters used in the examples were  $\alpha = 4.0$ ,  $\beta = 0.00001$ ,  $\gamma = 0.00001$ ,  $\omega = 4.0$ ,  $\rho = 4.0$  and  $\xi = 5.0$  and we selected them empirically.

Using *TAV* information from the extracted objects (figures 2(c) and 2(e)) we sampled the interpolated distance functions into three-dimensional volumes with  $50 \times 50 \times 70$  voxels. For normal extraction we had to select values for two parameters: maximum cluster angle constant  $\theta$  and sample displacement constant  $\delta$ . We chose  $\theta = 0.2$  radians empirically and  $\delta = 0.5$  voxel units as a negligible displacement value, taking into account that node positions have at least one voxel unit of separation between them (so accuracy is kept) and we are not interested in taking this value too low in order to avoid numerical errors in *natural coordinates* computation. For *natural neighbor interpolation* we kept  $\sigma = 1$ .

Figures 2(d) and 2(f) show the zero level isosurfaces for the femur and the tibia traced using Bloomenthal's polygonizer [2], while the six figures in last row of figure 2 are isolevel representations of sections traversing  $x$ ,  $y$  and  $z$  axes of the surface for the femur and tibia *implicit functions*. These images represent distance values in gray scale. Values in  $[0, \dots, 127]$  represent the inside (negative values) of the object and values in  $[128, \dots, 255]$  the outside. For clarity in representation, we insert a white level between adjacent levels inside the object, and a black level between adjacent levels outside.

## 5 Conclusions

In this work we apply *Topological Active Volumes (TAV)* for *CT* images segmentation. The *TAV* model has proved to give good results for this environment [1], but this time we have proved its usefulness for shape feature extraction using its topological information.

For the conversion of the *TAV* model into an implicit surface representation, we have used the *natural neighbor interpolation* based method proposed in [3], that guarantees correct results on a dense enough sample. *TAV* model produces sample points on the surface of the extracted objects and topological relations between them, that provide enough information to estimate normal direction and orientation. Using this information we analyse object surface and identify the areas where sampled density needs to be increased. Our preliminary results show the good performance of the method described.

## Acknowledgements

This paper has been partly funded by the Xunta de Galicia through the grant contracts PGIDIT03TIC10503PR, PGIDT04PXIC10501PN and PGIDIT04TIC206005PR.

## References

1. N. Barreira and M.G. Penedo. Topological Active Volumes for Segmentation and Shape Reconstruction of Medical Images. *Image Analysis and Recognition: Lecture Notes in Computer Science*, 3212:43–50, 2004.
2. J. Bloomenthal. An implicit surface polygonizer. In *Graphics gems IV*, pages 324–349. Academic Press Professional, Inc., 1994.
3. J.D. Boissonnat and F. Cazals. Smooth Surface Reconstruction via Natural Neighbor Interpolation of Distance Functions. *Proceedings of 16th Annual ACM Symposium on Computational Geometry*, pages 223–232, 2000.
4. D. E. Breen, S. Mauch, R. T. Whitaker, and J. Mao. 3D Metamorphosis between different types of geometric models. In *Eurographics 2001 Proceedings*, pages 36–48. Blackwell Publishers, September 2001.
5. R. O. Duda and P. E. Hart. *Pattern classification and scene analysis*. Wiley-Interscience, 1973.



6. V. Leboran, R. Dosil, and X. M. Pardo. Smooth Surface Reconstruction from Points and Normals using Implicit Surfaces. In *Actas del XIII Congreso Español de Informática Gráfica (CEIG'03)*, pages 203–216, 2003.
7. S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.
8. Chen S., J. F. O'Brien, and J. R. Shewchuk. Interpolating and Approximating Implicit Surfaces from Polygon Soup. In *Proc. of ACM SIGGRAPH 2004*, 2004.
9. R. Satherley and M. W. Jones. Hybrid distance field computation for volumetric objects. In *Proceedings of the Joint IEEE TCVG and Eurographics Workshop*, pages 121–133, 2001.
10. G. Yngve and G. Turk. Robust Creation of Implicit Surfaces from Polygonal Meshes. *IEEE Transactions on Visualization and Computer Graphics*, 8(4):346–359, 2002.

# Automatic Matching and Motion Estimation from Two Views of a Multiplane Scene\*

Gonzalo López-Nicolás, Carlos Sagüés, and José J. Guerrero

Dpto. de Informática e Ingeniería de Sistemas  
Instituto de Investigación en Ingeniería de Aragón, Univ. de Zaragoza  
Edificio Ada Byron, C/ María de Luna 1, E-50018 Zaragoza, Spain  
{gonlopez,csagues,jguerrer}@unizar.es

**Abstract.** This paper addresses the computation of motion between two views when 3D structure is unknown but planar surfaces can be assumed. We use points which are automatically matched in two steps. The first one is based on image parameters and the second one is based on the geometric constraint introduced by computed homographies. When two or more planes are observed, corresponding homographies can be computed and they can be used to obtain the fundamental matrix, which gives constraints for the whole scene. The computation of the camera motion can be carried out from a homography or from the fundamental matrix. Experimental results prove this approach to be robust and functional for real applications in man made environments.

**Keywords:** Matching points, multiplane scenes, homographies, fundamental matrix, motion estimation

## 1 Introduction

The fundamental matrix encapsulates the geometric information which relates two different views regardless of the observed scene. The non metric basis of this matrix makes possible to use uncalibrated cameras. It has been usually computed through points [1] although lines can also be used when two or more planes are available [2]. Obviously points can also be used to compute homographies and, if two or more homographies are available, the fundamental matrix can be computed from them [3], [4].

In all the cases the matching problem is crucial to make the process work automatically. The matching of features based on image parameters may give non matched or wrong matched features. Projective transformations allow image dependent measures, as cross-correlation, to be a viewpoint invariant, which make possible to afford wide baseline matching [5]. So, the constraint imposed by fundamental matrix or homographies must be used for matching points.

Scenes with several planes are usual in man made environments, and the model to work with multiple views of them is well known. Points or lines in one image of the world plane are mapped to points or lines in the other image by a

---

\* This work was supported by project DPI2003-07986.

plane to plane homography [6]. We robustly match points between two images using the projective transformations corresponding to the existing scene planes. The robust matching of points and the computation of the corresponding homography is iteratively carried out until we have no more available planes. If two planes have been computed at least, the fundamental matrix can be computed, which gives general constraint for the whole scene. It has been reported that the multi-plane algorithm is not as stable as the general method [3], but when less than three planes are observed, which is quite usual in man made environments, the multi-plane algorithm gives better results than the general method.

Camera motion between two views can be obtained from the computed homography or from the fundamental matrix. Both methods are exposed in this paper. Normally the computation of motion has been directly considered from the fundamental matrix, which is a more general model. However, the fundamental matrix is ill conditioned with short baseline or when all the points lie on a plane, which may easily happen in man made environments [6]. In these cases the fundamental matrix is an inappropriate model to compute camera motion. Using homographies, we can check the homology conditioning to determine if the fundamental matrix may be computed. Therefore we can choose the appropriate motion algorithm from either the fundamental matrix or the homography.

## 2 Robust Matching

Automatic matching continues to be an unsolved problem in general situations. The aim is to determine correspondences between points in two images without knowledge about motion or scene structure.

In this work the points of interest are extracted with the Harris corner extractor [7]. To obtain a homogeneous distribution of points all over the image, it is divided in a grid and we establish a maximum number of points per cell to be extracted. Additionally we establish a threshold of minimum contrast just to give only good points.

Later, we consider the matching in two steps, the first step is based on image correlation on a search window around the candidate points. This is actually the most weak step of our implementation because, as known, correlation is not invariant to rotations. As some mismatches appear here, we introduce in the second step, our "friendship" algorithm. It is similar to the previously proposed relaxation process [8]. The idea is to allow only the matches whose neighboring points move similarly. Those that do not behave as the neighbors are eliminated.

These points can be represented in the projective plane with homogeneous coordinates as  $\mathbf{p} = (x, y, 1)^T$ . A projective transformation  $\mathbf{H}_{21}$  exists from matched points belonging to a plane in such a way that  $\mathbf{p}_2 = \mathbf{H}_{21}\mathbf{p}_1$ .

From the previous relation each couple of corresponding points gives two homogeneous equations to compute the projective transformation, which can be determined up to a non-zero scale factor. To compute the homography, we have chosen the RANSAC method [9], which is a robust method to consider the existence of outliers. It makes a search in the space of solutions obtained from

subsets of four matches. Each subset provides a  $8 \times 9$  system of equations whose solution is obtained from singular value decomposition.

From here on, we introduce the geometrical constraint introduced by the estimated homography to get a bigger set of matches. Thus, final matches are composed by two sets. The first one is obtained from the matches selected after the robust computation of the homography. The second one is obtained making a rematching of not matched points based on the computed homography.

### 3 From Homographies to Fundamental Matrix

Fundamental matrix has been stated as a crucial tool when using uncalibrated images. As known, it is a  $3 \times 3$  matrix of rank 2 which encapsulates the epipolar geometry. It only depends on internal parameters of the camera and the relative motion.

Let us suppose the images are obtained with the same camera whose projection matrixes in a common reference system are  $\mathbf{P}_1 = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ ,  $\mathbf{P}_2 = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ ; being  $\mathbf{R}$  the camera rotation,  $\mathbf{t}$  the translation and  $\mathbf{K}$  the internal calibration matrix. Then, the fundamental matrix can be expressed as  $\mathbf{F}_{21} = \mathbf{K}^{-T}([\mathbf{t}]_{\times} \mathbf{R}) \mathbf{K}^{-1}$ . Normally, it has been computed from corresponding points [1], [10], using the epipolar constraint, which can be expressed as  $\mathbf{x}_2^T \mathbf{F}_{21} \mathbf{x}_1 = 0$ . However, the fundamental matrix is unstable when points lie in a plane [10]. In [3] is shown that the multiplane method behaves better than the general method when less than three planes are available. This constrained structure is usually observed in man made environments.

In the case of multiplane scenes some alternatives can be used to compute the fundamental matrix. If at least two homographies ( $\mathbf{H}_{21}^{\pi_1}, \mathbf{H}_{21}^{\pi_2}$ ) corresponding to two planes  $(\pi_1, \pi_2)$  can be computed between both images, the homology on the second image  $\mathbf{H}_2 = \mathbf{H}_{21}^{\pi_1} \cdot (\mathbf{H}_{21}^{\pi_2})^{-1}$ , which is a mapping from one image onto itself, can be computed. Under this mapping the epipole is a fixed point  $\mathbf{e}_2 = \mathbf{H}_2 \mathbf{e}_2$ , so it may be determined from the eigenvector of  $\mathbf{H}_2$  corresponding to non unary eigenvalue [6]. Therefore, the fundamental matrix can be computed using  $\mathbf{H}_{21}^{\pi_1}$  or  $\mathbf{H}_{21}^{\pi_2}$  as,

$$\mathbf{F}_{21} = [\mathbf{e}_2]_{\times} \mathbf{H}_{21}^{\pi_i} \quad , \quad (1)$$

being  $[\mathbf{e}_2]_{\times}$  the skew matrix corresponding to  $\mathbf{e}_2$  vector.

On the other hand, the fundamental matrix can also be computed from both homographies through a system of twelve linear equations extracted from the following relation [3],

$$\mathbf{H}_{21}^{\pi_i T} \mathbf{F}_{21} + \mathbf{F}_{21}^T \mathbf{H}_{21}^{\pi_i} = 0 \quad . \quad (2)$$

As we propose to compute fundamental matrix from homographies, a check on the homology conditioning may help to determine if the fundamental matrix may or may not be computed. Similarly the homology on the first image can be computed as  $\mathbf{H}_1 = (\mathbf{H}_{21}^{\pi_1})^{-1} \cdot \mathbf{H}_{21}^{\pi_2}$  and taking into account that for a plane  $\mathbf{H}_{21} = \mathbf{K}(\mathbf{R} - \frac{\mathbf{t} \mathbf{n}_{\pi}^T}{d_{\pi}}) \mathbf{K}^{-1}$ , it turns out that the eigenvalues of the  $\mathbf{H}_1$  homology are  $(1, 1, 1 + \mathbf{v}^T \mathbf{p})$  being  $\mathbf{v} = \mathbf{K} \mathbf{R}^{-1} \mathbf{t} / (1 - \frac{\mathbf{n}_{\pi_1}^T}{d_{\pi_1}} \mathbf{R}^{-1} \mathbf{t})$  a view dependent vector,

and  $\mathbf{p} = \left(\frac{\mathbf{n}_{\pi_1}^T}{d_{\pi_1}} - \frac{\mathbf{n}_{\pi_2}^T}{d_{\pi_2}}\right)\mathbf{K}^{-1}$  a plane dependent vector, being  $\mathbf{n}_{\pi_1}$ ,  $\mathbf{n}_{\pi_2}$  the normals and  $d_{\pi_1}$ ,  $d_{\pi_2}$  the distances of the planes [11].

So, the homology has two equal eigenvalues. The third one is related to the motion and the structure of the scene. These eigenvalues are used to test when two different planes have been computed, and then the epipole and the intersection of the planes can be also computed. The epipole is the eigenvector corresponding to the non-unary eigenvalue and the other two eigenvectors define the intersection line of the planes [6]. In case of small baseline or if there is only one plane in the scene, epipolar geometry is not defined and only one homography can be computed, so possible homology  $\mathbf{H}_1$  will be close to identity, up to scale.

In practice a filter is proposed using these ideas. Firstly, we normalize the homology dividing by the median eigenvalue. If there are no two unary eigenvalues, up to a threshold, then the computation is rejected. On the other hand, if the three eigenvalues are similar we check if the homology is close to identity to avoid the case where two similar homographies are computed.

## 4 Camera Motion from Two Views

Complete motion (rotation and translation up to a scale factor) can be computed from homography or from the fundamental matrix if camera is calibrated. As we have seen before, the homography  $\mathbf{H}_{21}$  can be related to motion in such a way that  $\mathbf{H}_{21} = \mathbf{K}(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^T}{d})\mathbf{K}^{-1}$ , being  $\mathbf{n}$  the normal to the scene plane and  $d$  its depth. From here, two solutions (up to a scale factor for  $\mathbf{t}$ ) can be obtained [12]. The main steps of this algorithm is summarized in Algorithm 1.

---

### Algorithm 1 Motion algorithm from homography

---

1. Compute a calibrated homography  $\mathbf{H}_{21}^c = \mathbf{K}^{-1}\mathbf{H}_{21}\mathbf{K}$
  2. Compute the singular value decomposition of matrix  $\mathbf{H}_{21}^c$ , in such a way that  $\mathbf{H}_{21}^c = \mathbf{U} \text{diag}(\lambda_1, \lambda_2, \lambda_3) \mathbf{V}^T$  with  $\lambda_2 = 1$
  3. Let be  $\mathbf{S}^T \mathbf{S} = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$ , and  $\alpha = \sqrt{\frac{\lambda_3 - \lambda_2}{\lambda_3 - \lambda_1}}$ ,  $\beta = \sqrt{\frac{\lambda_2 - \lambda_1}{\lambda_3 - \lambda_1}}$
  4. Writing  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ , compute  $\mathbf{v}_v = \alpha \mathbf{v}_1 + \beta \mathbf{v}_3$
  5. Compute rotation matrix  $\mathbf{R} = [\mathbf{H}_{21}^c \mathbf{v}_v, \mathbf{H}_{21}^c \mathbf{v}_2, \mathbf{H}_{21}^c \mathbf{v}_v \times \mathbf{H}_{21}^c \mathbf{v}_2][\mathbf{v}_v, \mathbf{v}_2, \mathbf{v}_v \times \mathbf{v}_2]^T$
  6. Compute translation up to a scale factor as  $\mathbf{t} = \mathbf{H}_{21}^c \mathbf{n} - \mathbf{R} \mathbf{n}$  being  $\mathbf{n} = \mathbf{v}_v \times \mathbf{v}_2$
  7. The second solution for  $\mathbf{R}$  and  $\mathbf{t}$  can be obtained by making  $\beta = -\beta$
  8. If  $\lambda_3 = \lambda_2$ , there is a sole solution being the camera translation perpendicular to the plane ( $\mathbf{t} \parallel \mathbf{R} \mathbf{n}$ ) and coming nearer the plane. If  $\lambda_1 = \lambda_2$  there is also a sole solution, but now the camera gets away from the plane. Finally, if  $\lambda_1 = \lambda_2 = \lambda_3$  report the sole solution  $\mathbf{t} = 0$ , and  $\mathbf{R} = \mathbf{H}_{21}^c$
- 

Camera motion can also be computed from the fundamental matrix. As in previous case, the algorithm provides two solutions up to a scale factor for translation. Given the calibration matrix, the motion can be deduced from  $\mathbf{F}$  as summarized in Algorithm 2 [6].

---

**Algorithm 2** Motion algorithm from fundamental matrix
 

---

1. Compute the essential matrix  $\mathbf{E} = \mathbf{K}^T \mathbf{F} \mathbf{K}$
  2. Compute the singular value decomposition of matrix  $\mathbf{E}$ , in such a way that  $\mathbf{E} = \mathbf{U} \text{diag}(1, 1, 0) \mathbf{V}^T$
  3. The camera translation, up to a scale factor is  $\mathbf{t} = \mathbf{U}(0, 0, 1)^T$
  4. The two solutions for the rotation matrix are  $\mathbf{R} = \mathbf{U} \mathbf{W} \mathbf{V}^T$  and  $\mathbf{R} = \mathbf{U} \mathbf{W}^T \mathbf{V}^T$ , being  $\mathbf{W} = [(0, 1, 0)^T, (-1, 0, 0)^T, (0, 0, 1)^T]$
- 

In case of pure rotation or if there exists only one plane in the scene, the epipolar geometry is not defined. Then, only the alternative of motion from homography will be correct.

## 5 Experimental Results

Many experiments have been carried out with synthetic and real images. The homology filter just commented has been used to determine when a second plane has been obtained. Several criteria can be used to measure the accuracy of the computed motion. With synthetic images, where motion is known, we measure the rotation error. We also measure the first order geometric error computed as the Sampson distance [6] for a set of corresponding points manually extracted and matched.

With real images the matches are automatically obtained for two planes in scene (Fig. 1). The points extracted are 479 from the first image and 475 from the second. The number of basic matches obtained is 147 with 86.4% of good matches. Once a homography has been computed, the robust homography computation and the growing matches process has been iteratively repeated twice. The experiment has been repeated 50 times using the same basic matches, and the mean of final matches obtained is 131.8 matches ( $\sigma = 10.5$ ) with 96.9% of good matches ( $\sigma = 1.2\%$ ). As it can be seen the number and quality of final matches are quite good.

As we have seen, one of the results of the homology is the intersection line of the planes. We have proposed to use a filter based on the homology eigenvalues to avoid situations where a sole homography can be computed or where the homographies do not give a right homology due to noise or bad extraction. In these cases the epipole, the fundamental matrix or the intersection line would be badly computed. In Fig. 2 we can see the intersection lines of the planes for 100 executions with and without the homology filter. As it can be seen the quality of the results improves significantly with the proposed filter.

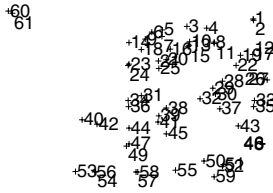
With respect to the fundamental matrix computation, we show (Table 1) the mean of the Sampson distance for 20 points manually extracted and matched. We consider the images of the college and two synthetic images. The synthetic scene consists of random points, with white noise of  $\sigma = 0.3$  pixels, distributed in three perpendicular planes. The experiment has been repeated 100 times and



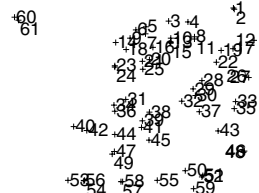
(a)



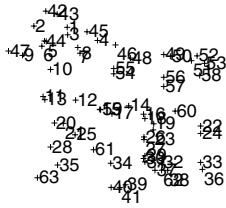
(b)



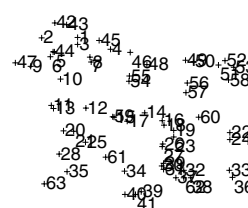
(c)



(d)



(e)

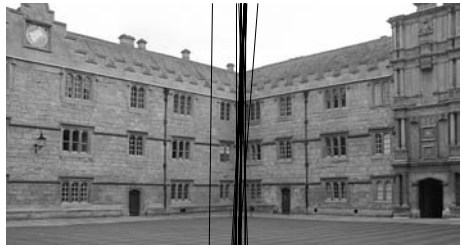


(f)

**Fig. 1.** Images of the college to compute homographies. Extracted points (a), (b). Matches corresponding to the first homography (c), (d) and to the second (e), (f). (Original images from VGG, Oxford)



(a)



(b)

**Fig. 2.** Intersection of the planes through the eigenvalues of the homology. The lines corresponding to 100 executions are represented without filter (a), and with homology filter (b)

**Table 1.** Sampson distance for 20 points manually matched (belonging to each plane for homographies and distributed around the scene for fundamental matrixes). We show in 100 executions the median and the mean with and without filter. These results are shown for the homographies (H1, H2) and for the fundamental matrixes: eH1 and eH2 using (1) with  $\mathbf{H}_{21}^{\pi_1}$  and  $\mathbf{H}_{21}^{\pi_2}$  respectively, and FH using (2)

		Synthetic (pixels)					Oxford college (pixels)				
		H1	H2	eH1	eH2	FH	H1	H2	eH1	eH2	FH
Without filter	median	0.581	0.586	0.891	0.789	0.932	0.707	0.683	1.004	1.286	1.906
	mean	0.577	0.586	1.619	1.458	1.634	0.709	0.698	4.998	5.187	12.61
With filter	median	0.581	0.584	0.740	0.725	0.805	0.687	0.666	0.566	0.796	1.045
	mean	0.578	0.587	0.926	0.767	0.883	0.697	0.694	0.642	0.789	1.099

we show mean and median values. The Sampson distance is similar for the three presented ways of computing the fundamental matrix, although it is a bit worse using (1). Probably this is because if one homography is less accurate than the other, (2) collects this inaccuracy, currently we are studying the implications of these differences.

**Table 2.** Mean of rotation error (Synthetic) and rotation angle (College) computing motion through homographies H1 or H2 with algorithm 1, and through fundamental matrixes, eH1 and eH2 using (1) and FH using (2), with algorithm 2

	Synthetic: rotation error (deg)					Oxford college: rotation (deg)				
	H1	H2	eH1	eH2	FH	H1	H2	eH1	eH2	FH
Without filter	0.958	0.454	0.524	0.545	0.562	9.240	10.64	7.777	7.662	8.096
With filter	0.456	0.365	0.225	0.226	0.214	9.691	10.97	9.118	9.115	9.478

Finally, results of the computation of camera motion using homographies and fundamental matrix are exposed. We have executed these algorithms 100 times. Table 2 shows the mean of the rotation (Oxford college) and the rotation error (synthetic data) obtained through homographies (Algorithm 1) and fundamental matrixes (Algorithm 2). Fundamental matrix is computed in different ways using equations (1) and (2). The results are exposed with and without the homology filter and they show the goodness of the proposed filter.

## 6 Conclusions

We have presented the matching of points, the computation of the intersection of the planes and the computation of camera motion from two views. This is carried



out through homographies corresponding to planes, which are quite usual in man made environments. The robust computation of matches based on homographies works especially well to automatically eliminate outliers which may appear when there is no information of scene structure or camera motion. The fundamental matrix and the intersection line of the planes is properly obtained if the images correspond to motion and scenes which are geometrically well conditioned. If it does not happen a homography may be given as a result of the algorithm and motion can be obtained from this homography.

The main achievement of this work is that all the process is made automatically and works in a robust way. Besides this, the joint use of homographies and fundamental matrix allows the proper selection of the model to determine camera motion in real applications. The proposed approach is a good solution in man made environments, where usually at least one plane is available.

## References

1. Zhang, Z., Deriche, R., Faugeras, O., Luong, Q.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence* **78** (1995) 87–119
2. Pellejero, O., Sagüés, C., Guerrero, J.: Automatic computation of fundamental matrix from matched lines. In: *Current Topics in Artificial Intelligence*, LNCS-LNAI 3040. (2004) 197–206
3. Luong, Q.T., Faugeras, O.: Determining the fundamental matrix with planes: Unstability and new algorithms. In: *Proc. Conference on Computer Vision and Pattern Recognition*, New-York (1993) 489–494
4. Rother, C., Carlsson, S.: Linear multi view reconstruction and camera recovery using a reference plane. *International Journal of Computer Vision* **49** (2002)
5. Pritchett, P., Zisserman, A.: Wide baseline stereo matching. In: *IEEE Conference on Computer Vision*. (1998) 754–760
6. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2000)
7. Harris, C., Stephens, M.: A combined corner and edge detector. The Plessey Company (1988)
8. Zhang, Z., Deriche, R., Faugeras, O., Luong, Q.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Rapport de recherche RR-2273, I.N.R.I.A., Sophia-Antipolis, France* (1994)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM* **24** (1981) 381–395
10. Luong, Q., Faugeras, O.: The fundamental matrix: Theory, algorithms, and stability analysis. *Int. Journal of Computer Vision* **17** (1996) 43–76
11. Zelnik-Manor, L., Irani, M.: Multiview constraints on homographies. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 214–223
12. Weng, J., Huang, T., Ahuja, N.: *Motion and Structure from Image Sequences*. Springer-Verlag, Berlin-Heidelberg (1993)

# Contextual Soccer Detection Using Mosaicing Techniques

Lluís Barceló and Xavier Binefa

Universitat Autònoma de Barcelona, UPIIA and Departament d'Informàtica  
Edifici Q, Bellaterra 08193, Barcelona, Spain

**Abstract.** Sport Video understanding aims to select and summarize important video events that occur in only special fragments of the whole sports video. A key aspect to this objective is to determine the position in the match field where the action takes place, that is, the location context of the play. In this paper we present a method to localize where in the match field the play is taking place. We apply our method to soccer videos, although the method is extensive to other sports. The method is based on constructing the mosaic of the first sequence that we process: this new mosaic is used as a *context mosaic*. Using this mosaic we register the frames of the other sequences in order to put in correspondence all the frames with the context mosaic, that is, put in context any play. In order to construct the mosaics, we have developed a novel method to register the soccer sequences based on tracking imaginary straight lines using the Lucas-Kanade feature tracker and the *vb-QMDPE* robust estimator.

## 1 Introduction

Distribution of sports video over various networks uses a high bandwidth and for this reason it is so difficult to find live sources of sports videos in the internet. However, processing sports sequences, for example detecting important events and creating summaries, allows to deliver sports videos even over narrow band networks or wireless, since the valuable semantics generally occupy only a small portion of the whole content.

It is also very important to index the content in order to make easy to search due to the ever growing size of content produced. For easy management a semantic index describing the different events in the content of the document is indispensable. Since manual annotation is unfeasible because of its tedious and cumbersome nature, automatic video indexing methods are necessary.

In literature several methods for automatic soccer analysis have been proposed, e.g. [1, 5, 9, 12]. One of the first reported methods was presented in [12]. The authors focus on visualization of ball and player tracks using mosaics. More recently, methods were proposed that try to narrow the semantic gap. In [1, 9] camera based detectors are proposed, exploiting the relation between the movement of the ball and the camera. A slow-motion replay detector is proposed in [5] as a strong indicator for an event of importance that happened beforehand. For

combination of the visual detectors a statistical Dynamic Bayesian Network is used in [1, 9], whereas [5] exploits a knowledge based approach.

In this paper we present a method to localize in soccer video sequences where in the match field the play is taking place. This kind of sequences are characterized by the fact that are generally very difficult to register because they contain a lot of moving objects (the players) and constant regions without texture or with a poor texture (low gradient) that correspond to the match field. For this reason we use a novel method in order to register the sequences based on tracking imaginary straight lines over the playfield. We want to find the *homography* that relates pairs of consecutive images, because the scene is planar (all match field are) and then the transformation that pass from one to other is a projective transformation (an homography). It is important to say that there are many classical methods like robust dense optical flow [2] and parametric methods [3] but with the presence of constant regions they do not work very fine.

Then we use this information in order to summarize the soccer sequences synthesizing an image mosaic. By computing the mosaics of different soccer sequences we can merge these mosaics in order to construct a larger mosaic that represents a wider area of the match field. Once we have done this wider version, we have each frame of each sequence in context, that is, localized in the larger mosaic. That is very important because in soccer sequences we lose sometimes the contextual reference because the camera focus a part of the field that does not contain the white lines of the field and therefore we do not have any contextual reference in order to know in which part of the field corresponds the frame.

The rest of the paper is organized as follows: in section 2 we explain the mosaic construction method, that includes our novel registration method and the synthesis of mosaics. Then in section 3 we present our contextual localization method and finally we present in section 4 the experimental results and in section 5 the conclusions and future work.

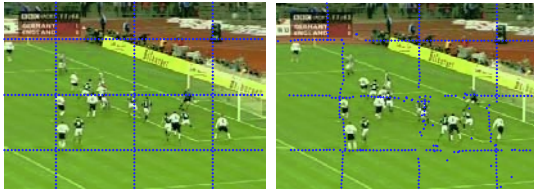
## 2 Mosaic Construction

Algorithms for the construction of image mosaics consist of two main steps: registration, i.e. estimating the transformations between every pair of consecutive frames of the video, and mosaic construction, i.e. the synthesis of the image mosaic from the previously estimated transformations and the frames of the video.

In our case in order to register the sequences we have developed a method based on tracking imaginary straight lines over the match field. In general terms, we have an initial set of features over the first frame and we track these features in the second frame. Then using the correspondences of these features we can extract the transformation that relates the two consecutive frames. In the next sections we explain in more detail the whole method.

### 2.1 Straight Lines Tracking

We have two consecutive images  $I_i$  and  $I_{i+1}$  and we want to obtain two corresponding sets of features that represents six imaginary straight lines. In order to



**Fig. 1.** Left: Initial features  $F$  in frame  $I_i$ . Right: The corresponding features of the left image,  $F'$ , in frame  $I_{i+1}$  using the *Pyramidal Lucas-Kanade Feature Tracker*. We can see that the estimations are affected by the moving objects (the players) and for the superimposed scoreboard.

do that, we select as initial features image points of  $I_i$  that represent six imaginary lines that must have the configuration shown in figure 1. These six lines are named as *control lines*. Therefore we have six straight lines  $R = [r_1, \dots, r_6]$  where each  $r_i = (a/c, b/c, 1)^T = (t, u, 1)^T$  corresponds to a straight line of equation:

$$ax + by + c = 0 \quad (1)$$

and we want to compute the homography that relates the transformation between the frame  $I_i$  and the frame  $I_{i+1}$ . It is known that, projective transformations keep straight lines [6]: so the corresponding features computed for each straight line in the frame  $I_i$  will also represent a straight line in the second frame  $I_{i+1}$ .

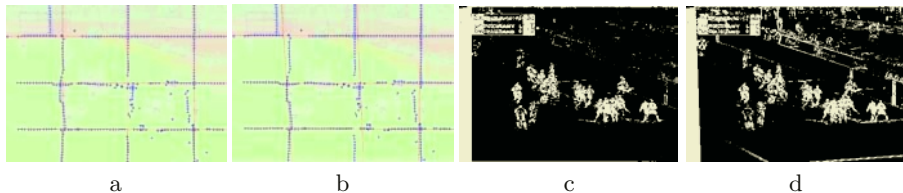
In order to compute the homography, we first compute a vector of features  $f_i$  for each straight line  $r_i$ , using as features, image points along of each line. As result, we have a set of features  $F = [f_1, \dots, f_6]$ , where each  $f_i$  is the corresponding vector of features of the straight line  $r_i$ .

Once we have the set of features of the frame  $I_i$ ,  $F$ , we want to find the respective features in the next frame  $I_{i+1}$ ,  $F'$ . Therefore, we apply a *Pyramidal Lucas-Kanade Feature Tracker* [4] to find this set of features. In figure 1 we can see the set of features  $F$  and its corresponding features  $F'$  after applying the Lucas-Kanade feature tracker in a soccer sequence.

Now, we have the set of features in the frame  $I_{i+1}$ ,  $F'$ , and we need a method to extract the six straight lines that best represent the set of features  $F'$ . However, the features contain a high percentage of outliers, in some cases more than 50% due to the moving objects (soccer players), but always there is a subset of good features. For this reason, we apply the *variable bandwidth QMDPE* robust estimator that is robust with more than 50% of outliers. In figure 2 we shown the estimation results of the vb-QMDPE, and we compare them with the Least-Square method. We can find detailed information about the variable bandwidth QMDPE robust estimator in [10, 11].

## 2.2 Homography Estimation

Once we obtain the respective six lines  $R' = [r'_1, \dots, r'_6]$  in the frame  $I_{i+1}$ , we know that a line  $r_i$  is transformed into  $r'_i$  using a projective transformation (homography) [6] in the following way:



**Fig. 2.** Images a) and b) contain the straight line estimation results using vb-QMDPE and Least-Squares methods respectively. The blue crosses are the features and the red lines the estimated straight lines. We can see that when we use the Least-Square method we obtain bad estimations due to the fact that we have a lot of outlier features. In contrast, when we use vb-QMDPE we obtain good estimations because this method is robust to more than 50% of outliers. In images c) and d), we show the registration errors using the previously estimations.

$$r'_i = (H^{-1})^T r_i \quad (2)$$

where  $r_i = (t, u, 1)^T$  and  $H$  is the homography represented by a non-singular  $3 \times 3$  matrix. Then, each line correspondence in the plane provides two equations in the 8 unknown entries of  $H$ . Therefore, it is necessary to find at least four line correspondences to define the transformation matrix uniquely, up to a scale factor. In our case we use six lines in order to make more robust the estimation because we deal with sequences that contain multiple moving objects. The equations of (2) can be rearranged in matrix form, obtaining the next system equation:

$$\begin{bmatrix} t'_i & 0 & -t_i t'_i & u'_i & 0 & -t_i u'_i & 1 & 0 \\ 0 & t'_i & -u_i t'_i & 0 & u'_i & -u_i u'_i & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} [h_{11} \ h_{12} \ h_{13} \ h_{21} \ h_{22} \ h_{23} \ h_{31} \ h_{32}]^T = \begin{bmatrix} t_i \\ u_i \\ \vdots \end{bmatrix} \quad (3)$$

and solving the above system equation using a *Least Squares* method we find the homography that relates the transformation between the frames  $I_i$  and  $I_{i+1}$ . Then we continue with the frames  $I_{i+1}$  and  $I_{i+2}$  using the previous method until we process the whole sequence.

### 2.3 Mosaic Synthesizing

Once we have processed the whole sequence we have the transformations between consecutive frames. However, in order to build the mosaic image we need that all the frames reference the same initial frame. For this reason, we calculate firstly the cumulative transformation of each frame with respect to the reference frame, in our case, the first frame of each sequence. We do that multiplying the transformation matrices to the left:



**Fig. 3.** Right: the context mosaic. Left: the mosaic of the play that we want to put in context.

$$\begin{aligned}
 H_{11} &= I_{3 \times 3} \\
 H_{12} &= H_{11}H_{12} \\
 &\vdots \\
 H_{1n} &= H_{11}H_{12}H_{23} \cdots H_{n-1n} = H_{1n-1}H_{n-1n}
 \end{aligned} \tag{4}$$

where  $H_{ij}$  is the homography between the frames  $I_j$  and  $I_i$  (the *cumulative transformation* between the frames). Now, in order to construct the final mosaic we have to transform each frame using its corresponding cumulative transformation and then we can apply a mean or median operator in order to obtain the mosaic using the whole transformed frames.

### 3 Contextual Localization

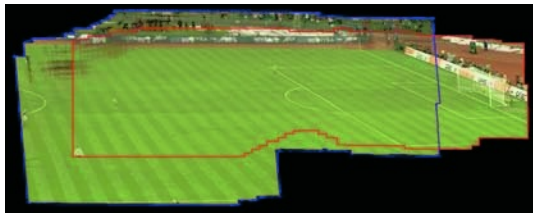
Now given a sequence we are able to construct its corresponding image mosaic using the method explained in previous sections. So, given the first sequence we construct its mosaic: this mosaic will be the *context* for the next sequences that we process and we name it as the *context mosaic*.

Then, in order to process the next sequences, we can use two methods: we can try to register the frames of the new sequence against the context mosaic, or we can build the mosaic image of the new sequence and then register this mosaic against the context mosaic. The first method is unfeasible because we could have a large transformation between the frame and the context mosaic, and moreover the frame could not contain contextual information (i.e. white lines of the match field) necessary to register against the context mosaic. For this reason we use the second method: we construct the mosaic of the new sequence and then we register both mosaics. Now, both mosaics contain contextual information and therefore the registration is feasible.

In figure 3 we can see both mosaics, the context mosaic and the mosaic of the new sequence. Both mosaics point to the same part of the field, but one is localized near of the center and the other is localized near of the penalty area.

#### 3.1 Initial Registration

We want to register both mosaics, but, first of all, a preliminary step is necessary in order to pre-register the mosaics because there is a huge transformation



**Fig. 4.** The synthesized mosaic using the mosaics of figure 3 after the initialization step (Hausdorff step) and the mosaic registration. We obtain a larger version that includes the two mosaics. As we can see it is not necessary that the context mosaic contains the mosaic of the sequence that we are processing.

between both mosaics. To do that we use the *Hausdorff Distance*. The Hausdorff Distance, given two finite point sets  $A = \{a_1, \dots, a_p\}$  and  $B = \{b_1, \dots, b_q\}$ , is defined as:

$$\begin{aligned}
 H(A, B) &= \max(h(A, B), h(B, A)) & (5) \\
 h(A, B) &= \max_{a \in A} \min_{b \in B} \|a - b\|
 \end{aligned}$$

where  $\|a - b\|$  is the  $L_2$  or Euclidean norm.

However, the Hausdorff distance measures the mismatch between two sets that are at *fixed* positions with respect to one another, whereas we are interested in comparing two images, where one of the images can be transformed by the action of some transformation group. Therefore, we use the *bidirectional Hausdorff distance* defined in [7, 8] to extract an initial registration of the two mosaics, that corresponds to a translation in  $x$  and  $y$  directions and a scale factor. We do not apply directly the Hausdorff distance to the mosaic images: we first apply a Discrete Laplacian to the mosaic images and then the Hausdorff distance.

### 3.2 Mosaic Registration

Using this initial estimation, we can register the two mosaics easily and obtain a mosaic image of mosaics as shown in figure 4. In this case we can use the traditional parametric methods or our method based on tracking imaginary lines, because both mosaics have a larger field of view and therefore are easy to register (once we have the pre-registration parameters).

Now, we have the cumulative transformations for each mosaic as explained in section 2.3, and the transformation between the mosaics. Then, in order to have the transformations that relates the frames of the sequences and the context mosaic, we have to obtain:

$$H_{total_n}^i = H_{seq_i} H_{1n}^i \quad (6)$$

where  $H_{1n}^i$  is the homography between the frames  $I_n$  and  $I_1$  of sequence  $i$  and  $H_{seq_i}$  is the homography that relates the transformation between the  $i$  sequence

mosaic and the context mosaic. Finally,  $H_{total_n}^i$  is the transformation between the frame  $I_n$  of sequence  $i$  and the context mosaic.

Once we have processed the new sequence we update the context mosaic using the mosaic of the new sequence (see mosaic of figure 4), and then we process the next sequence. We do that to extend the context mosaic with regions that it does not cover.

## 4 Experimental Results

Now, using all the extracted information we are able to recover the context of a frame that does not contain any information about its localization: this fact gives us a framework to the frame, that is, a context. In figure 5 we can see this concept graphically. Therefore, we have a compact representation of the sequences (the mosaic images) and the transformation between them.

Moreover, we can make a larger version of the sequences using the context mosaic as a background and superimposing the frames. This larger version of the sequences contains the same information but extended with more contextual information. We also could use the localization of the frames of a sequence in order to characterize the sequence as relevant or not in the soccer match.



**Fig. 5.** Left: a frame of the current sequence. This frame does not contain information about its localization over the match field as we can see. Right: the previous frame but with context, that is, printed over the context mosaic. This fact gives us information about the localization of the frame that with the single frame we do not have.

## 5 Conclusions and Future Work

We have presented a method to obtain the context of soccer sequences, that is, localize each frame of a sequence in the playfield. That fact allows us to obtain information very useful in order to index a soccer sequence or to characterize the sequence as relevant or not in the soccer match.

As a future work, it would be very interesting to use the transformation matrices  $H_{total_n}^i$  in order to construct a data structure that contains for each pixel of the context mosaic information about the frames that overlap this position. This data structure would make possible to index the sequence frames in function of their positions over the mosaic, and we can make searches like: *give me all the sequences that contain any frame in this position.*



## Acknowledgments

This work was funded by CICYT grant TIC2003-06075 and Generalitat de Catalunya grant 2001FI00060.

## References

1. J. Assfalg, M. Bertini, A. D. Bimbo, W. Nunziati, and P. Pala. Soccer highlights detection and recognition using hmms. *in Proc. IEEE Int. Conf. Multimedia and Expo*, 2002.
2. M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Fourth International Conf. on Computer Vision*, pages 231–236, 1993.
3. M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding: CVIU*, 63(1):75–104, 1996.
4. J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. *Microprocessor Research Labs*, 2000.
5. A. Ekin and A. M. Tekalp. Automatic soccer video analysis and summarization. *Symp. Electronic Imaging: Science and Technology: Storage and Retrieval for Image and Video Databases IV*, 2003.
6. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
7. D. P. Huttenlocher, G. A. Klenderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *Technical report TR 91-1211, Dept. of Computer science, Cornell University*, 1991.
8. D. P. Huttenlocher and W. J. Rucklidge. A multi-resolution technique for comparing images using the hausdorff distance. *Technical report TR 92-1321, Dept. of Computer science, Cornell University*, 1992.
9. R. Leonardi and P. Migliorati. Semantic indexing of multimedia documents. *IEEE Multimedia*, 9(2):44–51, 2002.
10. H. Wang and D. Suter. Variable bandwidth qmdpe and its application in robust optical flow estimation. In *ICCV03*, pages 178–183, Nice, France, 2003.
11. H. Wang and D. Suter. Mdpe: A very robust estimator for model fitting and range image segmentation. *International Journal of Computer Vision (IJCV)*, 2004.
12. D. Yow, B.-L. Yeo, M. Yeung, and B. Liu. Analysis and presentation of soccer highlights from digital video. *in Proc. Asian Conf. Computer Vision*, 1995.

# Probabilistic Image-Based Tracking: Improving Particle Filtering

Daniel Rowe, Ignasi Rius, Jordi Gonzàlez, Xavier Roca, and Juan J. Villanueva

Computer Vision Centre/Department of Computing Science  
Universitat Autònoma de Barcelona. 08193 Bellaterra, Barcelona, Spain  
drowe@cvc.uab.es

**Abstract.** *Condensation* is a widely-used tracking algorithm based on particle filters. Although some results have been achieved, it has several unpleasant behaviours. In this paper, we highlight these misbehaviours and propose two improvements. A new weight assignment, which avoids sample impoverishment, is presented. Subsequently, the prediction process is enhanced. The proposal has been successfully tested using synthetic data, which reproduces some of the main difficulties a tracker must deal with.

## 1 Introduction

The increasing interest in visual tracking is motivated by a huge number of promising applications that can now be tackled in real time thanks to recent technological advances. These applications include performance analysis, surveillance, video-indexing, smart interfaces, teleconferencing and video compression.

However, tracking agents can be extremely complex and time-consuming. To start with, strong requirements are mandatory. Real-time processing, extreme robust performances or high accuracy may be critical. On the other hand, difficulties common to all vision areas could cause system failures, specially in open environments. Hence, several of the following premises are often assumed: we can consider outdoors or indoors scenes, static or in-motion background, illumination changes, shadows, presence of clutter or a-priori known objects. Some foreground assumptions are also taken into account concerning whether a single or multiple agents should be expected; agents entries and exits from the scene; smooth, restricted or already-known dynamics; occlusions; carried objects; or appearance changes.

This paper focuses on solving some tracking problems related to the difficulties described above, such as multiple-agent tracking with unknown dynamics in presence of background clutter and strong noise. Specifically, we present some improvements to a well-known tracking algorithm, *Condensation* [3].

The remainder of this paper is organized as follows. Section 2 covers the probabilistic framework, revises *Condensation*, exposing its misbehaviours, and reviews a *Condensation*-based algorithm called *iTrack* [7]. Section 3 proposes several improvements on *Condensation/iTrack*. Section 4 shows experimental results with synthetic data and section 5 concludes this paper.

## 2 Image-Based Probabilistic Tracking

The **environment** is composed of agents, static objects and background conditions. The **scene** is defined as the piece of environment which a visual sensor can capture. The aim of the tracking task is to estimate the scene **state** over time. In this context, the state will be the parameterised knowledge which will characterise the scene evolution. Due to practical and theoretical ignorance, we do not have access to the ground truth. A probabilistic framework is commonly used as a way to perform tracking [5]. Classical approaches, such as the **Kalman Filter**, rely on linearity and gaussianity assumptions about the involved distributions. More recent works make use of **Bayesian filters** combined with **Monte Carlo Simulation** methods in order to deal with nonlinear and non-Gaussian transition models [1, 2]. Subsequent developments have introduced a re-sampling phase in the sequential simulation-based Bayesian filter algorithms. Such methods were first introduced in computer vision in *Condensation* [3]. However, they have several important drawbacks as stated in [4]. A great number of improvements have been introduced in recent years [6, 7] but there is still much ground to cover before solving unconstrained tracking.

### 2.1 Bayesian Filtering

The computation of the belief state  $\mathbf{S}_t$  given all evidence to date  $\mathbf{e}_{1:t}$  is called **filtering**. The posterior pdf<sup>1</sup> can be calculated through **recursive estimation**:

$$P(\mathbf{S}_t | \mathbf{e}_{1:t}) = \underbrace{P(\mathbf{e}_t | \mathbf{S}_t)}_{\text{likelihood}} \sum_{\mathbf{s}_{t-1}} \underbrace{P(\mathbf{S}_t | \mathbf{s}_{t-1})}_{\text{transition mod.}} \underbrace{p(\mathbf{s}_{t-1} | \mathbf{e}_{1:t-1})}_{\text{previous post.}} \quad (1)$$

updating
prediction

The pdf is projected forward according to the transition model, making a prediction, and it is updated in agreement with the likelihood function value based on the new evidence.

### 2.2 Condensation

Recursive estimation leads to expressions that are impossible to evaluate analytically unless strong assumptions are applied. *Condensation* addresses filtering when no assumption about linearity or gaussianity is made [3]. This problem is overcome by simulating  $N$  independent and identically-distributed samples from the posterior pdf,  $\{\mathbf{s}_t^i; i = 1 : N\}$ . The temporal prior  $\{\hat{\mathbf{s}}_t^i\}$  is obtained by applying the transition model to each sample. Weights  $\pi_t^i$  are assigned according to the likelihood function. Once all samples have been propagated and measured,

<sup>1</sup> Notation: bold case denotes vectors and matrices whereas non-bold case denotes scalars. Matrices are in uppercase. In a probabilistic context, uppercase denotes probability density functions (pdf) and random variables; lowercase denotes probabilities and variable instances.  $\mathbf{X}_{a:b}$  denotes a variable set from time  $t = a$  to  $t = b$ .

the set is re-sampled using normalized weights  $\bar{\pi}_t^i$  as probabilities. This sample set represents the new posterior. Expectations can be approximated as:

$$\mathbb{E}_{P(\mathbf{S}_t|e_{1:t})}(\mathbf{S}_t) \simeq \sum_{i=1}^N \bar{\pi}_t^i \hat{\mathbf{s}}_t^i = \frac{1}{N} \sum_{i=1}^N \mathbf{s}_t^i. \quad (2)$$

However, it has several unpleasant behaviours as stated in [4]. **Sampling impoverishment** is one of the main drawbacks of re-sampling algorithms. Samples are spread around several **modes** indicating hypotheses in the space state. Nevertheless, some of them are spurious. Similarly to genetic drift, there is a non-negligible probability of losing modes, a low probability of recovering them and the remaining modes could be all spurious. It can also be derived from this fact that different runs of the algorithm lead to different results. Therefore, computed expectations in different runs have high variance although computed expectations within the same algorithm run have low variance making the tracker look stable. On the other hand, *Condensation* has a tendency of clustering samples even when the likelihood function gives no information at all. In addition, the sample set size  $N$  is kept constant over time. Unfortunately, there is no information about how large  $N$  should be for a requested precision. Once  $N$  have been heuristically set, it may happen that at later times larger values of  $N$  may be required. Finally, *Condensation* was designed to keep multiple-hypothesis for a single agent.

### 2.3 iTrack

*iTrack* is a visual tracking algorithm based on *Condensation* [7], but both transition model and likelihood function are redefined. It also introduces some improvements in order to overcome some *Condensation* drawbacks and cope with multiple agents.

*iTrack* uses a first-order dynamic model in image coordinates to model the motion of the central point of a bounding box. The  $l$ -labeled agent's state is defined as  $\mathbf{s}_t^l = (\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t, \mathbf{A}_t)^T$  where each element represents the position, speed, bounding-box size and pixel appearance, respectively. The label associates one specific appearance model to the corresponding samples, allowing multiple-agent tracking. However, multiple-agent tracking causes several problems including that the agent with higher likelihood monopolizes the sample set. Denoting as  $N_j$  the number of samples belonging to the  $l$ -labeled agent, *iTrack* proposed the following normalization to avoid this issue:

$$\bar{\pi}_t^{i,l} = \frac{\pi_t^{i,l}}{\sum_{i=1}^N \pi_t^{i,j}} \frac{N_j}{N}, \quad \text{where } j = l. \quad (3)$$

An initial pdf, provided by a segmentation method, is needed to start the recursive estimation. *iTrack* also uses this pdf to reinitialize the algorithm allowing multiple-agent tracking and error recovery. Thus, some samples are generated according to the prior instead of being propagated.

### 3 Improving Condensation/iTrack

#### 3.1 Improvement 1. Sampling Impoverishment

Whether data association is feasible, using the prior density to generate new samples reduces the risk of sampling impoverishment. However, it is not completely avoided, since it depends on the probability of generating new samples, on whether these new samples represent the extinguishing mode, and on whether they can be associated to it. This problem is increased in a multiple-agent tracking scenario. Without considering new sample generation, losing an agent track is only a matter of time, according to the sample set size. In this case, those agents whose samples exhibit lower likelihood have higher probability of being lost, since the probability of propagating one mode is proportional to the cumulative weights of the samples that constitute it. Two kind of modes can be distinguished. In the first place, samples with different labels belong to different modes. Thus, several agents can be tracked simultaneously. Secondly, samples with the same label could be spread around different modes. This fact allows us to keep several hypotheses. Hopefully, one of them represents the true agent state and the others are due to background clutter.

In order to avoid single agent modes absorbing other agent samples, *genetic drift* must be prevented. This fact happens due to the lack of *genetic memory*: we propose to include a memory term which takes into account the number of agents being tracked. Hence, weights are normalized according to:

$$\bar{\pi}_t^{i,l} = \frac{\pi_t^{i,l}}{\sum_{j=1}^N \pi_t^{i,j}} \frac{1}{N_a}, \quad \text{where } j = l, \quad (4)$$

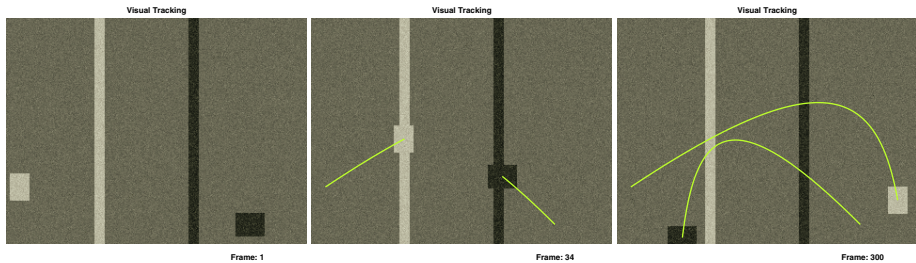
where  $N_a$  is the number of agents being tracked. It does not assign a fixed number of samples to each agent but ensures that each agent will have the same probability of being propagated. Furthermore, it can be combined with new sample generation, thereby improving the general performance. On the other hand, modes due to clutter are pruned because of differences in their dynamics. It is unlikely that any sample tracks local clutter since it implies highly abrupt changes in the dynamics. Non-losing the true mode depends on how accurate the dynamic model is, and how the different hypotheses are generated.

#### 3.2 Improvement 2. Agent Dynamics

*iTrack* makes predictions according to the following expressions:

$$\hat{\mathbf{x}}_t^i = \mathbf{x}_{t-1}^i + \mathbf{u}_{t-1}^i \Delta t + \xi_x^i, \quad \hat{\mathbf{u}}_t^i = \mathbf{u}_{t-1}^i + \xi_u^i. \quad (5)$$

The random terms  $\xi_x^i, \xi_u^i$  provide the system with a diversity of hypothesis. Samples with high likelihood are supposed to be propagated. Sample likelihoods depend on samples position but they do not depend on their speed. Thus, propagated samples could have an accurate position, but their speed values become


**Fig. 1.** Ground Truth

completely different from the agent's one in a few frames. Agents could be tracked since we are in a multiple-hypothesis scenario, but an important proportion of samples are wasted. The  $j$ -agent state is estimated according to:

$$\hat{\mathbf{s}}_t^j = \frac{1}{N_j} \sum_{i=1}^N \mathbf{s}_t^{i,j}. \quad (6)$$

Our approach proposes to feed-back the estimated agent speed at time  $t-1$ , denoted as  $\hat{\mathbf{u}}_{t-1}^j$ , into the prediction:

$$\hat{\mathbf{u}}_t^{i,j} = \hat{\mathbf{u}}_{t-1}^j + \xi_u^i. \quad (7)$$

However, there is still a weak relation between the agent and the estimated speeds: they are chosen only due to the sample weights, which do not depend on the current speed. We propose to enhance the estimation by considering not only the estimated speed from the selected samples but also by calculating the instant speed according to the history of positions. The following expressions update the agent position and speed recursively considering this fact:

$$\begin{aligned} \hat{\mathbf{x}}_t^j &= \hat{\mathbf{x}}_{t-1}^j (1 - \alpha_p) + \left( \frac{1}{N_j} \sum_{i=1}^N \mathbf{x}_t^{i,j} \right) \alpha_p, \\ \hat{\mathbf{u}}_t^j &= \hat{\mathbf{u}}_{t-1}^j (1 - \alpha_s) + \left( \hat{\mathbf{x}}_t^j - \hat{\mathbf{x}}_{t-1}^j \right) \alpha_s, \end{aligned} \quad (8)$$

where  $\alpha_p, \alpha_s$  denote the adaptation rates. The estimated speed is then fed-back when predicting the following sample state.

## 4 Experimental Results

In order to evaluate the algorithm performance, a two-moving-agent synthetic experiment has been designed. The aim is to cover several difficulties a tracker can run into, see Fig. 1. The background pixel intensity values are set randomly following a normal distribution. Both agents' pixel intensity values also have a normal distribution around different means. Two vertical strips are drawn in the background, simulating heavy clutter. Their distributions are identical to both agent's ones, thereby mimicking them. Strong acquisition-device noise, modeled

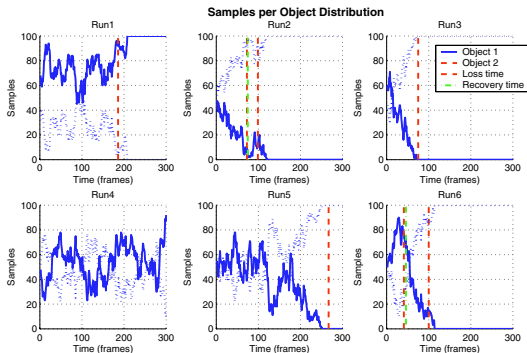


Fig. 2. Condensation/*iTrack* performance

Table 1. Performance of improvement 1 Table 2. Performance of improvement 1, 2

Mean normalized error		
	Agent 1	Agent 2
Run 1	0.1163	0.1309
Run 2	3.8864	0.1182
Run 3	0.1222	0.1226
Run 4	0.0980	0.1038
Run 5	0.1612	0.1131
Run 6	0.1101	2.4679

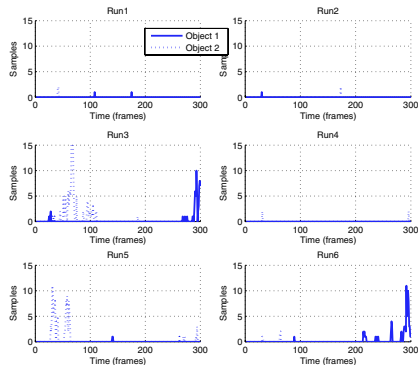
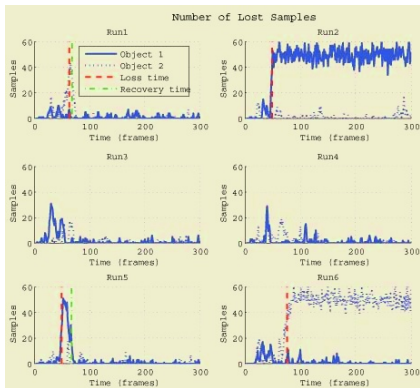
Mean normalized error		
	Agent 1	Agent 2
Run 1	0.0715	0.0716
Run 2	0.0849	0.1163
Run 3	0.0987	0.1289
Run 4	0.0645	0.0595
Run 5	0.0679	0.1173
Run 6	0.1233	0.0840

as White Additive Gaussian Noise, is simulated<sup>2</sup>. A highly non-linear dynamic is simulated: both agents move as projectiles which are shot into an environment with gravity and air friction. Tracking is performed over  $T = 300$  frames using  $N = 100$  samples. We present results of six random runs for each of the three approaches considered, namely, *iTrack* and both presented improvements. New sample generation is not used in order to evaluate only the tracking performance.

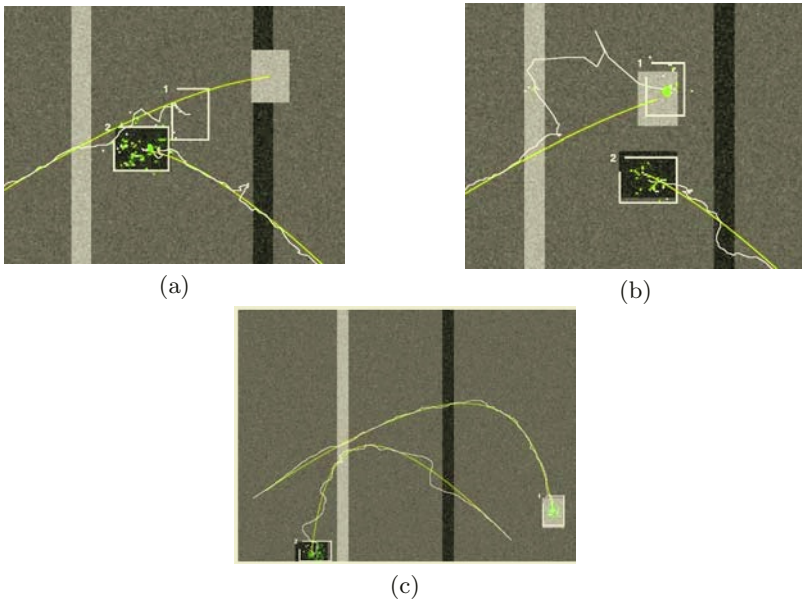
In 5 out of the 6 *iTrack* runs, an agent is lost due to the lack of samples, see Fig. 2. In the remaining one, at time  $t = 300$  an agent got 92% of the samples. An agent is considered lost when the normalized Euclidean distance, according to the agent size, between the agent and the estimation position is higher than a threshold set at 0.5. On the other hand, after the proposed weight normalization, the mean number of samples per agent fluctuates between 49.5 % and 50.5%.

Table 1 shows the mean normalized error, according to the agent size, in the estimation of the agent position before applying the new dynamics updating whereas Table. 2 shows the same results after applying it. A significant error reduction can be appreciated. Figs. 3, 4 compare the number of samples per agent that had lost the agent. After considering this improvement, a significant sample loss reduction is observed. Furthermore, none of the agents is ever lost.

<sup>2</sup> The standard deviation is set at 0.03 which implies nearly a ten per cent deviation.



**Fig. 3.** Performance of improvement 1 **Fig. 4.** Performance of improvements 1, 2  
(Notice that axes scale are reduced in 75%)



**Fig. 5.** Behaviour of the three studied trackers

The trackers behaviour can be seen in Fig. 5: Fig. 5.(a), corresponding to *iTrack*, shows how one of the agents absorbs all the samples. Fig. 5.(b), after applying the normalization improvement, shows agent recovery since the tracker have preserved enough samples to cope with multiple hypotheses. Thus, both modes, the agent and the clutter, are tracked until the clutter one disappears. Fig. 5.(c) shows the tracker performance once both improvements are considered.



## 5 Conclusions

In this paper, we have extended *Condensation* in order to enhance multiple-agent tracking. A new approach is taken to deal with one of *Condensation*'s great misbehaviours, the sampling impoverishment. This problem becomes critical in a multiple-tracking scenario. The new sample-weight normalization prevents from losing any of the targets due to the lack of samples. The dynamics updating is modified by feed-backing the estimated speed into the prediction stage. The agent speed is estimated combining two sources of knowledge: the fittest sample speed and the position historic. Thanks to both improvements, the tracker copes successfully with multiple-agent tracking. These agents have a highly non-linear dynamics which is successfully tracked using a constant-speed approach. Moreover, it also deals with complex clutter, which mimics the agent appearances, and strong noise. Improvements shown in these synthetic experiments are currently being applied in real applications relative to traffic surveillance. Encouraging results are being achieved.

## Acknowledgements

This work has been partially supported by the Spanish CICYT TIC 2003-08865.

## References

1. Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing* 50 (2): 174 - 188, 2002.
2. Doucet, A. On Sequential Simulation-Based Methods for Bayesian Filtering. CUED/F-INFENG/TR 310. University of Cambridge, 1998.
3. Isard, M. and Blake, A. Condensation - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision* 29 (1): 2 - 18, 1998.
4. King, O and Forsyth, D. A. How Does Condensation Behave with a Finite Number of Samples? *ECCV proceedings* (1): 695 - 709, 2000.
5. Russell, R. and Norvig, P. *Artificial Intelligence, a Modern Approach*. Chapters 13-15. Prentice Hall, 2003.
6. Merwe, R. van der, Doucet, A., de Freitas, N. and Wan, E. The Unscented Particle Filter. CUED/F-INFENG/TR 380. University of Cambridge, 2000.
7. Varona, X., Gonzàlez, J., Roca, X. and Villanueva, J. J. *iTrack: Image-based Probabilistic Tracking of People*. *ICPR* (3): 7122 - 7125, 2000.

# A Framework to Integrate Particle Filters for Robust Tracking in Non-stationary Environments

Francesc Moreno-Noguer and Alberto Sanfeliu

Institut de Robòtica i Informàtica Industrial, UPC-CSIC  
Llorens Artigas 4-6, 08028, Barcelona, Spain  
{fmoreno, asanfeliu}@iri.upc.es

**Abstract.** In this paper we propose a new framework to integrate several particle filters, in order to obtain a robust tracking system able to cope with abrupt changes of illumination and position of the target. The proposed method is analytically justified and allows to build a tracking procedure that adapts online and simultaneously the colorspace where the image points are represented, the color distributions of the object and background and the contour of the object.

## 1 Introduction

The integration of several visual features has been commonly used to improve the performance of tracking algorithms [1, 3, 9, 10]. However, all these methods lack a robust dynamic model to track the state of the features and cope with abrupt and unexpected changes of the target's position or appearance. Particle filters have been demonstrated to be robust enough to track complex dynamics. Usually, particle filters have been applied to only one object feature. [4] tracks an object based on multiple hypotheses of its contour. Subsequently, several approaches [7, 8] predict the target position based on the particle filter formulation. In our previous work [6] we proposed the use of this framework to predict the object and background color distributions.

In this work, we introduce a framework for the integration of several particle filters which are not independent between them, so that we can fuse their respective predicted features. [5] integrates different particle filter algorithms for tracking tasks, but with the assumption that the algorithms are conditionally independent. That is, if particle filter  $\mathcal{PF}_1$  is based on features  $\mathbf{z}_1$  to estimate the state vector  $\mathbf{x}_1$  and particle filter  $\mathcal{PF}_2$  uses features  $\mathbf{z}_2$  to estimate  $\mathbf{x}_2$ , for each whole state of the object  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2\}$  it is assumed that,  $p(\mathbf{z}_1, \mathbf{z}_2 | \mathbf{X}) = p(\mathbf{z}_1 | \mathbf{x}_1)p(\mathbf{z}_2 | \mathbf{x}_2)$ . But this assumption is very restrictive and many times is not satisfied. For instance, a usual method to weigh each one of the samples of a contour particle filter, is based on the ratio of the number of pixels inside the contour with object color versus the number of pixels outside the contour with background color. This means that the contour feature is not independent of the color feature. In this situation if  $\mathbf{z}_1$  represents the color features and  $\mathbf{z}_2$  the contour ones, the latter will be function of both  $\mathbf{x}_1$  and  $\mathbf{z}_1$ , i.e.  $\mathbf{z}_2 = \mathbf{z}_2(\mathbf{x}_1, \mathbf{z}_1)$ . Previous equation should be rewritten as,  $p(\mathbf{z}_1, \mathbf{z}_2 | \mathbf{X}) = p(\mathbf{z}_1 | \mathbf{x}_1)p(\mathbf{z}_2 | \mathbf{z}_1, \mathbf{x}_1, \mathbf{x}_2)$ . In this paper we will design a system that verifies this relation of dependence between object features. The main contributions of the paper are the following:

1. Proposal of a framework to integrate several conditionally dependent particle filters.
2. There is no restriction in the number of particle filters that can be integrated.
3. Use the method to develop a robust tracking system that: **(a)** Adapts online the color space where image points are represented. **(b)** Adapts the distributions of the object and background colorpoints. **(c)** Accommodates the contour of the object.

All these features make our system capable to track objects in complex situations, like unexpected changes of the scene color, or abrupt and non-rigid movements of the target, as will be shown in the results Section.

In Section 2 we will introduce the mathematical framework and analytical justification of the method. The features that will be used to represent the object are described in Section 3. In Section 4 we will depict details about the sequential integration procedure for the real tracking. Results and conclusions will be given in Sections 5 and 6.

## 2 Mathematical Framework

In the general case, let's describe the object being tracked by a set of  $F$  features,  $\mathbf{z}_1, \dots, \mathbf{z}_F$ , that are sequentially conditional dependent, i.e. feature  $i$  depends on feature  $i - 1$ . Each one of these features is associated to a state vector  $\mathbf{x}_1, \dots, \mathbf{x}_F$ , which conditional a posteriori probability  $p_1 = p(\mathbf{x}_1|\mathbf{z}_1), \dots, p_F = p(\mathbf{x}_F|\mathbf{z}_F)$  is estimated using a corresponding particle filter  $\mathcal{P}\mathcal{F}_1, \dots, \mathcal{P}\mathcal{F}_F$ . For the whole set of variables we assume that the dependence is only in one direction:

$$\{\mathbf{z}_k = \mathbf{z}_k(\mathbf{z}_i, \mathbf{x}_i), \mathbf{x}_k = \mathbf{x}_k(\mathbf{x}_i, \mathbf{z}_i)\} \iff i < k \quad (1)$$

Considering this relation of dependence we can add extra terms to the a posteriori probability computed for each particle filter. In particular, the expression for the a posteriori probability computed by  $\mathcal{P}\mathcal{F}_i$  will be  $p_i = p(\mathbf{x}_i|\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{z}_1, \dots, \mathbf{z}_i)$ . Keeping this in mind, next we will prove that the whole a posteriori probability can be computed sequentially, as follows:

$$\begin{aligned} P &= p(\mathbf{X}|\mathbf{Z}) = p(\mathbf{x}_1, \dots, \mathbf{x}_F|\mathbf{z}_1, \dots, \mathbf{z}_F) \\ &= p(\mathbf{x}_1|\mathbf{z}_1)p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2) \cdots p(\mathbf{x}_F|\mathbf{x}_1, \dots, \mathbf{x}_{F-1}, \mathbf{z}_1, \dots, \mathbf{z}_F) = p_1 p_2 \cdots p_F \end{aligned} \quad (2)$$

*Proof.* We will prove this by induction, and applying Bayes' rule [2] and Eq. 1:

– Proof for 2 features:

$$p(\mathbf{x}_1, \mathbf{x}_2|\mathbf{z}_1, \mathbf{z}_2) = p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)p(\mathbf{x}_1|\mathbf{z}_1, \mathbf{z}_2) = p(\mathbf{x}_1|\mathbf{z}_1)p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$$

– For  $F - 1$  features we assume that

$$\begin{aligned} p(\mathbf{x}_1, \dots, \mathbf{x}_{F-1}|\mathbf{z}_1, \dots, \mathbf{z}_{F-1}) &= (\mathbf{x}_1|\mathbf{z}_1)p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2) \\ &\cdots p(\mathbf{x}_{F-1}|\mathbf{x}_1, \dots, \mathbf{x}_{F-2}, \mathbf{z}_1, \dots, \mathbf{z}_{F-1}) \end{aligned} \quad (3)$$

– Proof for  $F$  features:

$$\begin{aligned} p(\mathbf{x}_1, \dots, \mathbf{x}_F|\mathbf{z}_1, \dots, \mathbf{z}_F) &= p(\mathbf{x}_F|\mathbf{x}_1, \dots, \mathbf{x}_{F-1}, \mathbf{z}_1, \dots, \mathbf{z}_F)p(\mathbf{x}_1, \dots, \mathbf{x}_{F-1}|\mathbf{z}_1, \dots, \mathbf{z}_{F-1}) \\ &\stackrel{\text{Eq. 3}}{=} p(\mathbf{x}_1|\mathbf{z}_1)p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2) \cdots p(\mathbf{x}_F|\mathbf{x}_1, \dots, \mathbf{x}_{F-1}, \mathbf{z}_1, \dots, \mathbf{z}_F) \end{aligned}$$

Eq.2 tells us that the whole a posteriori probability density function can be computed sequentially, starting with  $\mathcal{PF}_1$  to generate  $p(\mathbf{x}_1|\mathbf{z}_1)$  and use this to estimate  $p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$  with  $\mathcal{PF}_2$ , and so on.

In the iterative performance of the method,  $\mathcal{PF}_i$  also receives as input at iteration  $t$ , the output *pdf* of its state vector  $\mathbf{x}_i$  at the iteration  $t - 1$ . We write the time expanded version of the *pdf* for  $\mathcal{PF}_i$  as  $p_i^{(t)} = p(\mathbf{x}_i^{(t)}|\mathbf{x}_1^{(t)}, \dots, \mathbf{x}_{i-1}^{(t)}, \mathbf{z}_1^{(t)}, \dots, \mathbf{z}_i^{(t)}, p_i^{(t-1)})$ . We can also expand the expression of the whole *pdf* from Eq.2 as follows:

$$\begin{aligned} P^{(t)} &= p(\mathbf{X}^{(t)}|\mathbf{Z}^{(t)}) = p(\mathbf{x}_1^{(t)}, \dots, \mathbf{x}_F^{(t)}|\mathbf{z}_1^{(t)}, \dots, \mathbf{z}_F^{(t)}) \\ &= p(\mathbf{x}_1^{(t)}|\mathbf{z}_1^{(t)}, p_1^{(t-1)}) \cdots p(\mathbf{x}_F^{(t)}|\mathbf{x}_1^{(t)}, \dots, \mathbf{x}_{F-1}^{(t)}, \mathbf{z}_1^{(t)}, \dots, \mathbf{z}_F^{(t)}, p_F^{(t-1)}) \\ &= p_1^{(t)} p_2^{(t)} \cdots p_F^{(t)} \end{aligned}$$

Now let's describe in some detail the updating procedure of the  $i - th$  particle filter,  $\mathcal{PF}_i$ . At time  $t$ , the filter receives  $p_i^{(t-1)}$ , the *pdf* of the state vector  $\mathbf{x}_i$  at time  $t - 1$ . This distribution is approximated by a set of samples  $\mathbf{s}_{ij}^{(t-1)}$ ,  $j = 1 \dots N_i$ , with associated weights  $\pi_{ij}^{(t-1)}$ . Given the set  $\{\mathbf{s}_{ij}^{(t-1)}, \pi_{ij}^{(t-1)}\}$  the value of  $p_i^{(t)}$  is estimated using the standard particle filter procedure:

1. The set  $\{\mathbf{s}_{ij}^{(t-1)}, \pi_{ij}^{(t-1)}\}$ ,  $j = 1 \dots N_i$  is resampled (sampling with replacement) according to the weights  $\pi_{ij}^{(t-1)}$ . We obtain the new set  $\{\mathbf{s}'_{ij}{}^{(t-1)}, \pi_{ij}^{(t-1)}\}$ .
2. Particles  $\mathbf{s}'_{ij}{}^{(t-1)}$  are propagated to the new set  $\{\mathbf{s}_{ij}^{(t)}\}$ ,  $j = 1 \dots N_i$ , based on the random dynamic model  $\mathbf{s}_{i,j}^{(t)} = \mathcal{H}_i \mathbf{s}'_{i,j}{}^{(t-1)} + \mathbf{p}_i$ , where  $\mathcal{H}_i \sim \mathcal{A}_{3 \times 3}(0, \sigma_{H_i})$  and  $\mathbf{p}_i \sim \mathcal{T}_{3 \times 1}(\mu_{p_i}, \sigma_{p_i})$ . We define the matrix  $\mathcal{A}$  and the vector  $\mathcal{T}$  as follows:

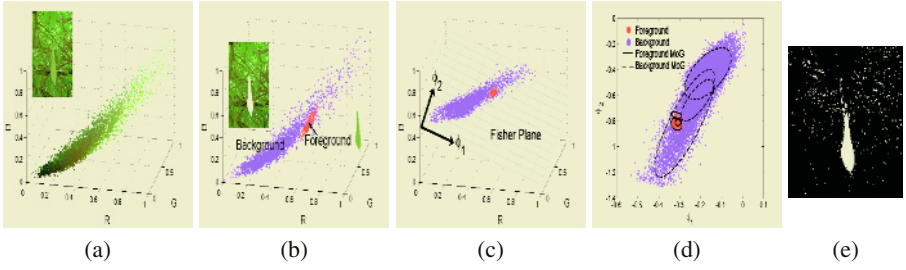
$$\mathcal{A}_{m \times m}(\mu_A, \sigma_A) = \begin{bmatrix} 1 + a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & 1 + a_{mm} \end{bmatrix} \quad \mathcal{T}_{m \times 1}(\mu_t, \sigma_t) = [t_1, \dots, t_m]^T \quad (4)$$

where  $a_{ij} \sim \mathcal{N}(\mu_{A_{ij}}, \sigma_{A_{ij}})$ ,  $t_i \sim \mathcal{N}(\mu_{t_i}, \sigma_{t_i})$ .

3. Finally, using some external measure on the feature  $\mathbf{z}_i^{(t)}$  (updated with the values of the set of features  $\{\mathbf{z}_k^{(t)}\}$ ,  $k < i$  and its corresponding state vectors  $\{\mathbf{x}_k^{(t)}\}$ ), samples  $\mathbf{s}_{ij}^{(t)}$  are weighted in order to obtain the output of iteration  $t$ , that is  $\{\mathbf{s}_{ij}^{(t)}, \pi_{ij}^{(t)}\}$ ,  $j = 1 \dots N_i$ , approximating  $p_i^{(t)}$ .

### 3 Features Used for a Robust Tracking

In order to design a system able to work in real and dynamic environments we define a set of features that include both appearance (normal direction of the Fisher plane [6] and the color distribution of the object) and geometric attributes (contour) of the object. Next we will describe each one of these features:



**Fig. 1.** Color model. (a) All image points in the RGB colorspace. In the upper left part the original image is shown. (b) Manual classification of image points in foreground ( $\mathcal{O}$ ) and background ( $\mathcal{B}$ ). (c) Projection of  $\mathcal{O}$  and  $\mathcal{B}$  points on the Fisher plane. (d) *MoG* of  $\mathcal{O}$  (the central leaf) and  $\mathcal{B}$  in the Fisher colorspace. (e)  $p(\mathcal{O}|\mathbf{c}^{Fisher})$ , where brighter points correspond to more likely pixels.

### 3.1 Normal to the Fisher Plane

In [6] we first introduced the concept of Fisher colorspace, and suggested that for tracking purposes the best colorspace is one that maximizes the distance between the object and background colorpoints. Let the sets  $\mathcal{C}_O^{RGB} = \{\mathbf{c}_{O,i}^{RGB}\}$ ,  $i = 1, \dots, N_O$  and  $\mathcal{C}_B^{RGB} = \{\mathbf{c}_{B,j}^{RGB}\}$ ,  $j = 1, \dots, N_B$  be the colorpoints of the object and background respectively, represented in the 3-dimensional RGB colorspace.

Fisher plane  $\Phi = [\phi_1, \phi_2] \in \mathcal{M}_{3 \times 2}$  is computed applying the nonparametric Linear Discriminant Analysis technique [2] over the sets  $\mathcal{C}_O^{RGB}$  and  $\mathcal{C}_B^{RGB}$ . An RGB colorpoint  $\mathbf{c}^{RGB}$  is transformed to the 2D Fisher colorspace by  $\mathbf{c}^{Fisher} = \Phi^T \mathbf{c}^{RGB}$  (see Fig. 1). This colorspace is adapted online, through the particle filter formulation presented above, with a 3D state vector corresponding to its normal vector,  $\mathbf{x}_1 = \phi_1 \times \phi_2$ .

### 3.2 Color Distribution of the Foreground and Background

In order to represent the color distribution of the foreground and background in the Fisher colorspace, we use a *mixture of gaussians (MoG)* model. The conditional probability for a pixel  $\mathbf{c}^{Fisher}$  belonging to a multi-colored object  $\mathcal{O}$  is expressed as a sum of  $M_o$  gaussian components:  $p(\mathbf{c}^{Fisher}|\mathcal{O}) = \sum_{j=1}^{M_o} p(\mathbf{c}^{Fisher}|j) P(j)$ . Similarly, the background color will be represented by a mixture of  $M_b$  gaussians. Given the foreground ( $\mathcal{O}$ ) and background ( $\mathcal{B}$ ) classes, the a posteriori probability that a pixel  $\mathbf{c}^{Fisher}$  belongs to object  $\mathcal{O}$  is computed using the Bayes rule (Fig. 1d,e):

$$p(\mathcal{O}|\mathbf{c}^{Fisher}) = \frac{p(\mathbf{c}^{Fisher}|\mathcal{O}) P(\mathcal{O})}{p(\mathbf{c}^{Fisher}|\mathcal{O}) P(\mathcal{O}) + p(\mathbf{c}^{Fisher}|\mathcal{B}) P(\mathcal{B})} \quad (5)$$

where  $P(\mathcal{O})$ ,  $P(\mathcal{B})$  are the a priori probabilities of  $\mathcal{O}$  and  $\mathcal{B}$ .

The configurations of the *MoG* for  $\mathcal{O}$  and  $\mathcal{B}$  are parameterized by the vector  $\mathcal{G}_\varepsilon = [\mathbf{p}_\varepsilon, \mu_\varepsilon, \lambda_\varepsilon, \theta_\varepsilon]$  where  $\varepsilon = \{\mathcal{O}, \mathcal{B}\}$ ,  $\mathbf{p}_\varepsilon$  contains the priors for each gaussian component,  $\mu_\varepsilon$  the centroids,  $\lambda_\varepsilon$  the eigenvalues of the principal directions and  $\theta_\varepsilon$  the angles between the principal directions and the horizontal.  $\mathbf{x}_2 = \{\mathcal{G}_O, \mathcal{G}_B\}$  will be the state vector representing the color model.

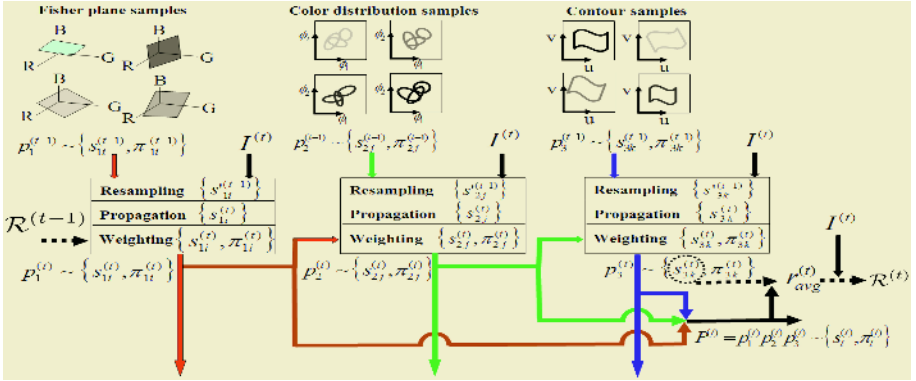


Fig. 2. Flow diagram of one iteration of the complete algorithm.

### 3.3 Contour of the Object

Since color segmentation usually gives a rough estimation about the object location, we use the contour of the object, to obtain a more precise tracking. The contour will be represented by  $N_c$  points in the image,  $\mathbf{r} = [(u_1, v_1), \dots, (u_{N_c}, v_{N_c})]^T$ . We assign these values to the state vector,  $\mathbf{x}_3 = \mathbf{r}$ .

## 4 The Complete Tracking Algorithm

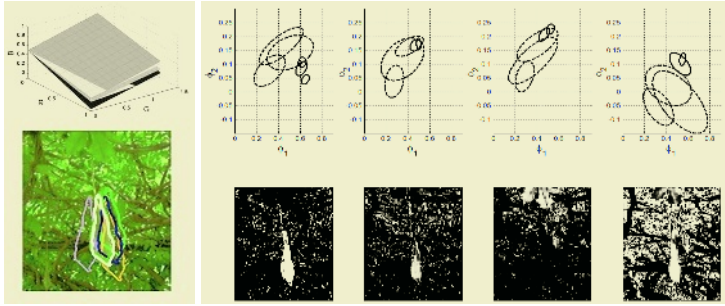
In this Section we will integrate the tools described previously and analyze the complete method for tracking rigid and non-rigid objects in cluttered environments, under changing illumination. Let's describe the algorithm step by step (See Fig. 2):

### 4.1 Input at Iteration $t$

At time  $t$ , for each  $i$ -feature,  $i = 1, \dots, 3$ , a set of  $N_i$  samples  $\mathbf{s}_{ij}^{(t-1)}$ ,  $j = 1, \dots, N_i$  (with the same structure than  $\mathbf{x}_i$ ), is available from the previous iteration. Each sample has an associated weight  $\pi_{ij}^{(t-1)}$ . The whole set represents an approximation the a posteriori *pdf* of the system,  $P^{(t-1)} = p(\mathbf{X}^{(t-1)} | \mathbf{Z}^{(t-1)})$ , where  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$  contains the state vectors, and  $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$  refers to the measured features. Also available is the set of image points  $\mathcal{R}^{(t-1)}$  that discretizes the contour of the object, and the input RGB image at time  $t$ ,  $I^{RGB,(t)}$ .

### 4.2 Updating the Fisher Plane *pdf*

At the starting point of iteration  $t$ ,  $\mathcal{PF}_1$ , the particle filter associated to  $\mathbf{x}_1$ , receives at its input  $P_1^{(t-1)}$ , the *pdf* of the state vector  $\mathbf{x}_1$  at time  $t - 1$ , approximated with  $N_1$  weighted samples  $\{\mathbf{s}_{1j}^{(t-1)}, \pi_{1j}^{(t-1)}\}$ ,  $j = 1, \dots, N_1$ . These particles are resampled and propagated to the set  $\{\mathbf{s}_{1j}^{(t)}\}$  according to the dynamic model. Each sample represents a



**Fig. 3.** Generation of multiple hypotheses for each feature. Upper left: Fisher plane. Lower left: Contour of the object. Right: Color distributions (and the corresponding a posteriori *pdfs* maps).

different Fisher plane,  $\Phi_j, j = 1, \dots, N_1$ . In order to assign a weight to each propagated sample, we define a region  $W$  in the image  $I^{RGB,(t)}$ , where we expect the object will be (bounding box around the contour  $\mathcal{R}^{(t-1)}$ ). We fit a *MoG* configuration to the points inside and outside  $W$ , and assign a weight to each Fisher plane  $\Phi_j$  depending on how well it discriminates the two regions:

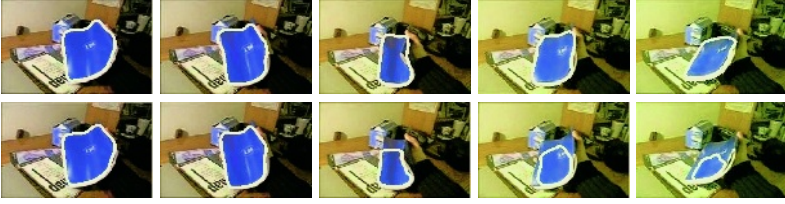
$$\pi_{1j}^{(t)} \sim \frac{1}{N_W} \sum_{(u,v) \in W} p\left(W|I(u,v)_j^{Fisher,(t)}\right) - \frac{1}{N_{\overline{W}}} \sum_{(u,v) \notin W} p\left(W|I(u,v)_j^{Fisher,(t)}\right) \quad (6)$$

where  $I_j^{Fisher,(t)}$  is the image  $I^{RGB,(t)}$  projected on the plane  $\Phi_j$ , and  $N_W, N_{\overline{W}}$  are the number of image pixels in and out of  $W$ , respectively.

### 4.3 Updating the Foreground and Background Color Distributions *pdfs*

$\mathcal{PF}_2$ , the particle filter associated to the state vector  $\mathbf{x}_2$ , receives at its input  $p_2^{(t-1)} \sim \{s_{2j}^{(t-1)}, \pi_{2j}^{(t-1)}\}, j = 1, \dots, N_2$ , approximating the *pdf* of the color distributions in the previous iteration, and  $p_1^{(t)} \sim \{s_{1k}^{(t)}, \pi_{1k}^{(t)}\}, k = 1, \dots, N_1$ , an approximation to the *pdf* of the Fisher planes at time  $t$ . Particles  $\{s_{2j}^{(t-1)}\}$  are resampled and propagated (using the dynamic model associated to  $\mathbf{x}_2$ ) to the set  $\{s_{2j}^{(t)}\}$ . A sample  $s_{2j}^{(t)}$  represents a *MoG* configuration for the foreground and background. For the weighting stage, we associate to this sample, a sample of Fisher plane from  $\mathcal{PF}_1$ , in such a way that those samples  $s_{1k}^{(t)}$  of Fisher planes having higher probabilities will be assigned more times to the samples  $s_{2j}^{(t)}$  of *MoGs*. The weighting function is similar as before, but now the *MoGs* are provided by the sample  $s_{2j}^{(t)}$ .

$$\pi_{2j}^{(t)} \sim \frac{1}{N_W} \sum_{(u,v) \in W} p\left(\mathcal{O}|I(u,v)_j^{Fisher,(t)}\right) - \frac{1}{N_{\overline{W}}} \sum_{(u,v) \notin W} p\left(\mathcal{O}|I(u,v)_j^{Fisher,(t)}\right) \quad (7)$$



**Fig. 4.** Tracking results of a bending book in a sequence with smooth lighting changes. Upper row: using the proposed method the tracking works. Lower row: using only a contour particle filter and assuming smooth change of color the method fails.

#### 4.4 Updating the Contour *pdf*

$\mathcal{PF}_3$ , receives at its input  $p_3^{(t-1)} \sim \{\mathbf{s}_{3j}^{(t-1)}, \pi_{3j}^{(t-1)}\}$ ,  $j = 1, \dots, N_3$ , that approximates the *pdf* of the contours in the previous iteration, and  $p_2^{(t)} \sim \{\mathbf{s}_{2k}^{(t)}, \pi_{2k}^{(t)}\}$ ,  $k = 1, \dots, N_2$ , an approximation to the *pdf* of the color distributions of foreground and background at time  $t$ . The set  $\{\mathbf{s}_{3j}^{(t)}\}$  (the resampled and propagated particles, see Fig. 3) are weighted based on  $p_2^{(t)}$  through a similar process than described for  $\mathcal{PF}_2$ : first we associate a sample  $\mathbf{s}_{2k}^{(t)}$  to each sample  $\mathbf{s}_{3j}^{(t)}$ , according to the weight  $\pi_{2k}^{(t)}$ . Then we use the a posteriori probability map  $p(\mathcal{O}|I_j^{Fisher, (t)})$  assigned to  $\mathbf{s}_{2k}^{(t)}$  in the previous step, and the contour  $\mathbf{r}_j$  represented by  $\mathbf{s}_{3j}^{(t)}$  to compute the weight as follows:

$$\pi_{3j}^{(t)} \sim \frac{1}{N_{\mathbf{r}_j}} \sum_{(u,v) \in \mathbf{r}_j} p(\mathcal{O}|I(u,v)^{Fisher, (t)}) - \frac{1}{N_{\overline{\mathbf{r}_j}}} \sum_{(u,v) \notin \mathbf{r}_j} p(\mathcal{O}|I(u,v)^{Fisher, (t)}) \quad (8)$$

where  $N_{\mathbf{r}_j}$  and  $N_{\overline{\mathbf{r}_j}}$  are the number of image pixels inside and outside the contour  $\mathbf{r}_j$ .

The whole *pdf* can be approximated by a set of samples and weights:

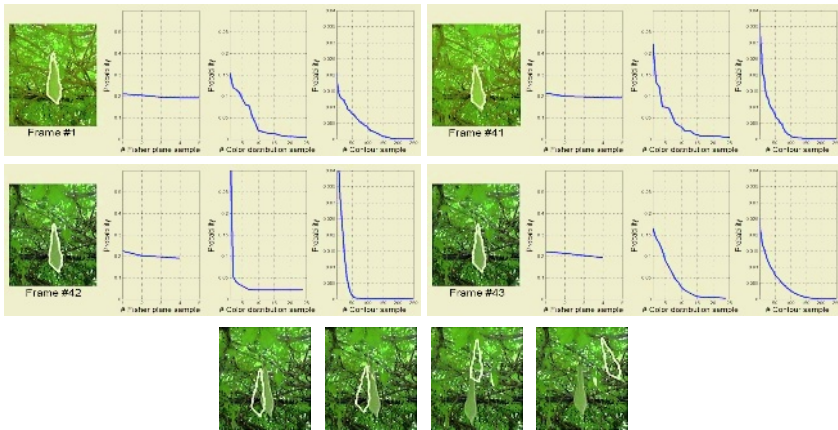
$$P^{(t)} = P^{(t)}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 | \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) = p_1^{(t)} p_2^{(t)} p_3^{(t)} \sim \{s_l^{(t)}, \pi_l^{(t)}\} \quad l = 1, \dots, N_3 \quad (9)$$

Considering these final weights, the output contour is computed as  $\mathcal{R}^{(t)} = \sum_{l=1}^{N_3} s_{3l}^{(t)} \pi_l^{(t)}$ .

## 5 Experimental Results

In this Section we examine the robustness of our system to several changing conditions of the environment, in situation where other algorithms may fail. In the first experiment we track the boundary of a bending book in a video sequence, where the lighting conditions change smoothly from natural lighting to yellow lighting. The upper row of Fig. 4 shows some frames of the tracked results. The same video sequence is processed by a particle filter that only uses multihypotheses for the prediction of the contour feature, while the color is predicted using a smooth dynamic model. Lower row of Fig. 4 shows that this method is unable to track the contour of the object and cope with the effects of self-shadowing produced during the movement of the book.





**Fig. 5.** Tracking results of a cluttered sequence with abrupt change of illumination and unpredictable movement of the target. Up: Results using the proposed method, and weight distribution for each particle filter. Down: Results assuming smooth change of color.

In the second experiment we have tested the algorithm with a sequence of a moving leaf. Although this is a challenging sequence because it is highly cluttered, the illumination changes abruptly and the target moves unpredictably, the tracking results using the proposed method are good. Upper images of Fig. 5 show some frames of the tracking results. We show also the distribution of the weights for the samples of each particle filter. Observe that during the abrupt change of illumination (between frames 41 and 42), there is a compression of these curves. This means that the number of samples predicted well has been reduced. Nevertheless, the difference of probability between these samples and the rest of the samples has increased meaning that in next iteration the new predictions will be centered on these ‘good’ particles. We can observe that for frame 43 the tracking has stabilized. On the other hand, the lower images of Fig. 5 show the inability to accommodate these abrupt changes using a contour particle filter with smooth color prediction.

## 6 Conclusions

In this paper we have presented a new technique to integrate different particle filters that are conditionally dependent. This framework has allowed us to design a tracking algorithm that accommodates simultaneously the colorspace where the image points are represented, the color distributions of the object and background and the contour of the object. We have demonstrated the effectiveness of the method both analytically and experimentally, tracking real sequences presenting high content of clutter, non-rigid objects, non-expected target movements and abrupt changes of illumination.

## Acknowledgements

This work was supported by CICYT projects DPI2001-2223 and DPI2000-1352-C02-01, and by a fellowship from the Spanish Ministry of Science and Technology.

## References

1. S.Birchfield, "Elliptical head tracking using intensity gradients and color histograms", *Proc. CVPR*, pp.232-237, 1998.
2. K.Fukunaga, "Introduction to statistical pattern recognition", 2nd ed., *Academic Press*, 1990.
3. E.Hayman, J.O.Eklundh, "Probabilistic and voting approaches to cue integration for figure-ground segmentation", *Proc. ECCV*, pp.469-486, 2002.
4. M.Isard, A.Blake, "CONDENSATION-Conditional Density Propagation for visual tracking", *IJCV*, Vol.29(1), pp.5-28, 1998.
5. I.Leichter, M.Lindenbaum, E.Rivlin, "A probabilistic framework for combining tracking algorithms", *Proc. CVPR*, Vol.2, pp.445-451, 2004.
6. F.Moreno-Noguer, A.Sanfeliu, D.Samaras, "Fusion of a Multiple Hypotheses Color Model and Deformable Contours for Figure Ground Segmentation in Dynamic Environments", *Proc. ANM, CVPRw*, 2004.
7. K.Nummiaro, E.Koller-Meier, L.Van Gool, "An adaptive color-based particle filter", *IVC*, Vol.2(1), pp.99-110, 2003.
8. H.Sidenbladh, M.J.Black, D.J.Fleet, "Stochastic tracking of 3D human figures using 2D image motion", *Proc. ECCV*, pp.702-718, 2000.
9. M.Spengler, B.Schiele, "Towards robust multi-cue integration for visual tracking", *Machine Vision and Applications*, Vol. 14(1), pp. 50-58, 2003
10. J.Triesch, C.von der Malsburg, "Democratic integration: self-organized integration of adaptive cues", *Neural Computation*, Vol.13(9), pp.2049-2074, 2001.

# Stereo Reconstruction of a Submerged Scene<sup>\*</sup>

Ricardo Ferreira<sup>1</sup>, João P. Costeira<sup>1</sup>, and João A. Santos<sup>2</sup>

<sup>1</sup> Instituto de Sistemas e Robótica / Instituto Superior Técnico

<sup>2</sup> Laboratório Nacional de Engenharia Civil

**Abstract.** This article presents work dedicated to the study of refraction effects between two media in stereo reconstruction of a tridimensional scene. This refraction induces nonlinear effects making the stereo processing highly complex. We propose a linear approximation which maps this problem into a new problem with a conventional solution. We present results taken both from synthetic images generated by a raytracer and results from real life scenes.

## 1 Introduction

Physical modelling is, still today, the main tool for testing and designing costal structures, specially rubble-mound breakwaters. One of the most important failure modes of this kind of structure is the armour layer hydraulic instability caused by wave action. Currently, to test the resistance of a proposed design to



**Fig. 1.** Real and model breakwater.

this failure mode, a scale model of the structure is built in a wave tank or in a wave flume, such as the one shown in figure (1), and it is exposed to a sequence of surface waves that are generated by a wave paddle. One of the parameters that have proved of paramount importance in the forecast of the structure behaviour is the profile erosion relative to the initial undamaged profile. Thus, measuring and detecting changes in the structure's envelope is of paramount importance.

<sup>\*</sup> This work was supported by the Portuguese FCT POSI programme under framework QCA III and project MEDIRES of the AdI.

Laser range finders are one obvious and easy way of reconstructing the scene, however, since common lasers do not propagate in the water, the tank (or flume) have to be emptied every time a measurement is taken.

This is a quite expensive procedure, both in time and money resources. We propose to use a stereo mechanism to reconstruct a submersed scene captured from cameras placed outside of the water. This way we can monitor both the emerged and submerged part of the breakwater.

## 1.1 Problem Definition

The problem tackled in this article is the reconstruction of a 3D scene with a stereo pair. Between the scene and the cameras there is an interface that bends light rays according to Snell's law.

The main difficulty here is that the known epipolar constraint, which helps reducing the search for a match, is not usable. Unlike conventional wisdom, straight lines underwater do not project as straight lines in the image. As figure 1.c illustrates, for each pixel in one image, possible matches are along a curve which is different for every point on the object. Essentially, this means that most stereo algorithms are unusable. We show that, if the incidence angle is small, the linear part of the Taylor Series expansion, which is equivalent to modifying camera parameters, is precise enough for our purpose. In other words current stereo algorithms can be used, provided the camera orientation parameters are within a certain range.

Though with a relatively straightforward solution, to our knowledge, this problem has not been addressed in the literature since most systems are placed underwater, thus eliminating the refraction issue.

## 2 Scene Reconstruction in the Presence of an Interface

### 2.1 Snell's Law

Whenever an interface is involved, Willebrord Snell's Law will necessarily be spoken of. The law states that a light ray crossing an interface will be bent according to

$$k_1 \sin \varphi_i = k_2 \sin \varphi_r$$

where  $\varphi_i$  and  $\varphi_r$  are the angles the incident and refracted light rays have with respect to the normal of interface at the point of intersection. Considering a planar interface at  $z = 0$  (see figure 1), a light ray emitted from a point above the interface will relate to its refracted ray by:

$$\begin{aligned} v_r^x(\mathbf{v}_i) &= \frac{k_1}{k_2} v_i^x, & v_r^y(\mathbf{v}_i) &= \frac{k_1}{k_2} v_i^y \\ v_r^z(\mathbf{v}_i) &= -\sqrt{\left(1 - \frac{(k_1)^2}{(k_2)^2}\right) \left((v_i^x)^2 + (v_i^y)^2\right) + (v_i^z)^2}. \end{aligned} \quad (1)$$

This non-linear relation can be simplified by expanding  $v_r^z(\mathbf{v}_i)$  in its Taylor series (in the neighborhood of  $\mathbf{v}_i = [0\ 0\ -1]^T$ ) and retaining the first order term. This results in a much simpler (linear) transformation

$$\mathbf{v}_r \approx \begin{bmatrix} kv_i^x \\ kv_i^y \\ v_i^z \end{bmatrix} = \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{v}_i, \quad \text{where } k = \frac{k_1}{k_2}. \quad (2)$$

## 2.2 Image Rectification

This approximation leads to a simple image rectification process, cancelling most of the distortion introduced by the interface. Using equation (2) and classic geometry, it can be shown that all light rays converge at a single point  $\mathbf{p}_1$ , as illustrated in figure 2. The relation between both focal points is done by:

$$\mathbf{p}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{k} \end{bmatrix} \mathbf{p}_2. \quad (3)$$

This fact hints at the possibility of rectifying the image with refraction effects by only changing the extrinsic camera parameters. In other words, by approximating Snell's law, the problem with refraction is transformed into a typical stereo problem “without” air-water interface. All that remains to be done is to project the original image onto the  $z = 0$  plane, and project it back to a virtual camera with projection center at  $\mathbf{p}_1$ . If  $\mathcal{P}_2$  and  $\mathcal{P}_1$  are, respectively, the original camera projection matrix and the virtual camera projection matrix, the rectification consists of a homography, given by:

$$\mathbf{H} = \mathcal{P}_1 \mathbf{M}(\mathbf{p}_2) \mathcal{P}_2^*. \quad (4)$$

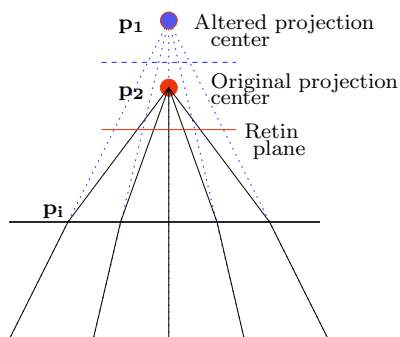
Here, the operator  $\{\cdot\}^*$  denotes matrix pseudo-inverse which projects a point in image coordinates onto the camera projection plane (at  $z = 1$  in camera coordinates). Matrix  $\mathbf{M}(\mathbf{p}_2)$  projects a point onto the  $z = 0$  plane using  $\bar{\mathbf{p}}_2$  as a projection center. It is defined by:

$$\mathbf{M}(\mathbf{p}_2) = \begin{bmatrix} -p_2^z & 0 & p_2^x & 0 \\ 0 & -p_2^z & p_2^y & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -p_2^z \end{bmatrix}. \quad (5)$$

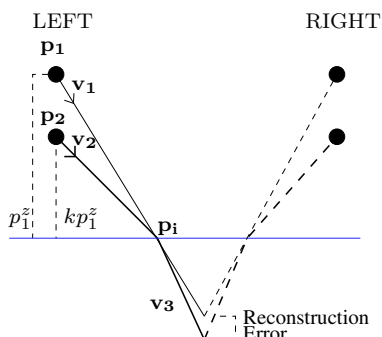
The intrinsic parameters of the virtual camera are chosen to minimize information loss or any other criteria needed by the specific implementation. In particular in the case of stereo reconstruction, the image rectification process imposes a few constraints on these parameters.

## 2.3 Underwater Stereo Reconstruction

The previous rectification process changes the image in such a way that they become suitable to classic stereo reconstruction algorithms. Be advised though



**Fig. 2.** Representation of the path followed by a beam of light when the first order snell approximation is used.



**Fig. 3.** Illustration of the correction needed to Snell's equations after image rectification.

that no guarantee was made about epipolar lines. Generally, depending on the resolution used, baseline, and angle of incidence of the light rays, the epipolar constraint does not occur due to the effect of higher order terms, neglected by the Snell rectification. In case the rectification mentioned above is not accurate enough, two dimensional search must be done to match the images. In these circumstances, rectification can significantly narrow the band of search around the estimated epipolar line.

Although the matching process gains considerably by assuming the simplification as valid, for greater reconstruction precision the nonlinear terms shouldn't be discarded. After the matching has been done, the true Snell deformation can be taken into account. In other words, equations 1 must be modified to include the rectification effect on the image coordinates. This is illustrated in figure 3. Note that  $\mathbf{v}_3$  is the true trajectory of the underwater light beam and not  $\mathbf{v}_1$ . We know how to obtain  $\mathbf{v}_3$  from  $\mathbf{v}_2$ , but now only  $\mathbf{v}_1$  is available. Finding the intersection of the line through  $\mathbf{p}_1$  tangent to  $\mathbf{v}_1$  with the plane  $z = 0$  yields  $\mathbf{p}_i$

$$\mathbf{p}_i = \left[ p_1^x - \frac{p_1^z}{v_1^z} v_1^x \quad p_1^y - \frac{p_1^z}{v_1^z} v_1^y \quad 0 \right]^T. \quad (6)$$

As mentioned before, Snell's approximation changed the camera's focal point. Knowledge about the original camera's focal point ( $\mathbf{p}_2$ ) allows us to find  $\mathbf{v}_2$ :

$$\mathbf{p}_2 = [p_1^x \quad p_1^y \quad kp_1^z]^T, \quad \mathbf{v}_2 = \mathbf{p}_i - \mathbf{p}_2 = \left[ -\frac{p_1^z}{v_1^z} v_1^x \quad -\frac{p_1^z}{v_1^z} v_1^y \quad -kp_1^z \right]^T.$$

Replacing this expression of  $\mathbf{v}_2$  in equation 1, we can represent  $\mathbf{v}_3$  exclusively as a function of the virtual camera, that is:

$$\mathbf{v}_3 \propto \left[ v_1^x \quad v_1^y \quad -\sqrt{\frac{1-k^2}{k^2} \left( (v_1^x)^2 + (v_1^y)^2 \right) + (v_1^z)^2} \right]^T. \quad (7)$$

It is now possible to apply equations (6) and (7) to the left and right cameras to triangulate for the 3D point. Due to the discrete nature of the sensors the two lines do not usually intersect, so a least squares error approach is used.

## 2.4 Implementation Notes

The location of the water plane is obtained during the calibration process using a floating checkered board. For a description on how to use this plane to calibrate the cameras' extrinsic (and intrinsic) parameters please see Bouguet's work [2] which is based on Zhang [3] and Heikkilä [4]. As stated before, the water plane is forced (calibrated) to be at  $z = 0$ . In order to facilitate point matching, the calibration data is then used to project the left and right images on a common plane making the epipolar lines horizontal [5]. These images are then processed by any classic stereo reconstruction algorithm. In our case we were interested in a dense stereo reconstruction so we used Sun's algorithm [6] based on dynamic programming.

Please note that what is discussed in this paper is valid only for underwater scenes. If the scene to be reconstructed is only partially submerged, two reconstructions should be performed. One valid for all the pixels corresponding to points over water, and another for the pixels corresponding to underwater points. Since the water plane is at  $z = 0$ , it can be written as  $\mathbf{w} = [0 \ 0 \ 1 \ 0]^T$  in projective coordinates. This plane can be easily described in disparity space as  $\mathbf{w}_d = H^{-T} \mathbf{w}$ , using the projective transformation

$$\mathbf{H} = \mathcal{D}\mathcal{E}, \quad \text{where} \quad \mathcal{D} = \begin{bmatrix} f & 0 & c_i^x & 0 \\ 0 & f & c_i^y & 0 \\ 0 & 0 & c_r^x - c_i^x & -Bf \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

$\mathcal{E}$  is the world to camera projective transformation and  $\mathcal{D}$  is the camera to d-space transformation with  $f$  describing the focal length,  $c_i^j$  the  $j$  coordinate ( $x$  or  $y$ ) of the principal point of camera  $i$  (left or right) and  $B$  is the baseline between left and right cameras (see for example [7]). It is then possible to know in a disparity map which camera pixels correspond to points under or above water.

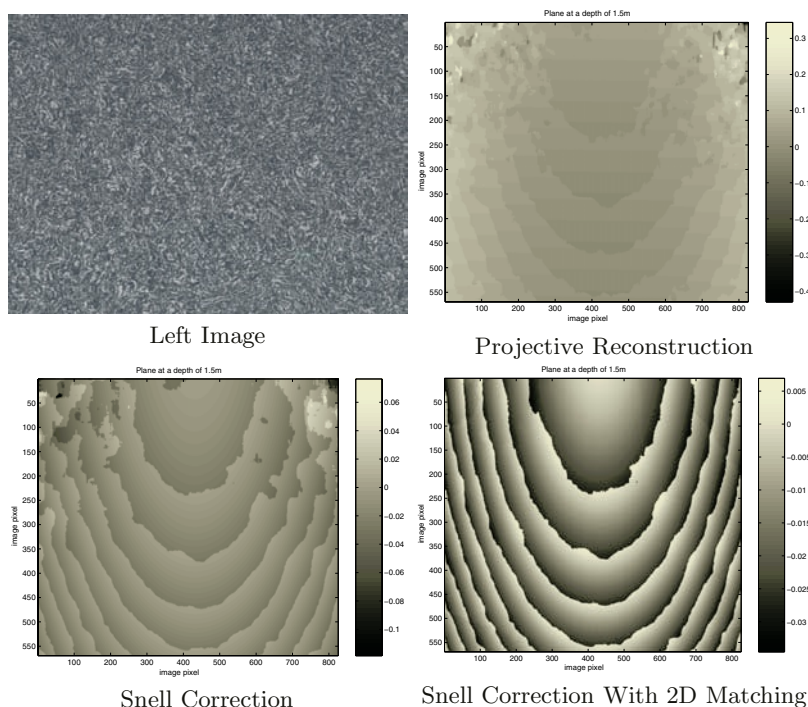
## 3 Experiments

To validate the algorithm, two different experiments were made. First a synthetic scene with planes at different depths was created. Images rendered from this scene are completely known to us, allowing reconstruction errors to be measured. The second type of images are real world images from a model breakwater. Since we do not have "ground truth" we can evaluate performance only qualitatively.

### 3.1 Synthetic Experiment

A few synthetic images were generated using povray<sup>1</sup> consisting of textured planes at various depths. The cameras are placed at 1.3m over the interface

<sup>1</sup> One of the oldest raytracers still used, which correctly models refraction effects.



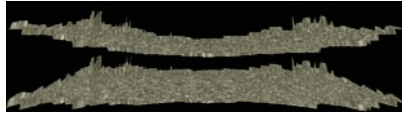
**Fig. 4.** Reconstruction error in depth (meters) for each pixel. The reconstructed scene consists of a textured plane at a depth of 1.5m as illustrated in the first image.

(looking slightly away from the perpendicular) with a baseline of 25cm . Please note that all of these reconstructions assume that the epipolar constraint is valid. This is clear in all the plane images since the matching algorithm starts to fail when the incidence angle becomes too great (noticeable in the top corners of the error images).

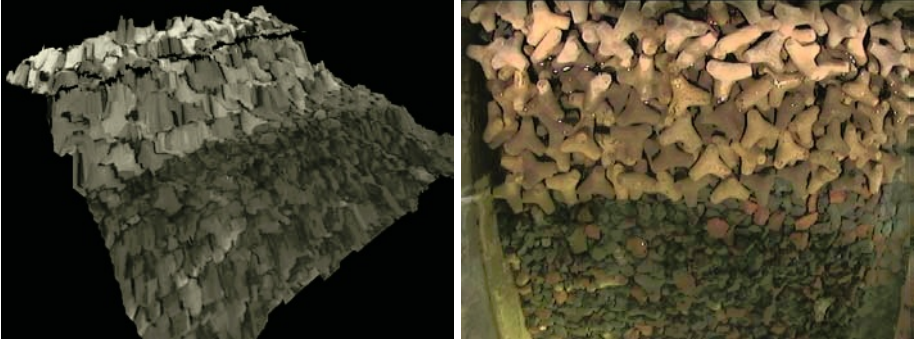
The first error image shown in figure 4 describes the reconstruction error when it is assumed that the disparity space is a projective reconstruction of the scenery. Note that Snell approximation is still used to help feature matching. The plane is reconstructed as a paraboloid (barely noticeable in the error images) due to the fact that higher order terms of Snell’s law are discarded. This effect is much clearer in figure 5 where the actual plane reconstruction is shown. The top corners of the error image are poorly reconstructed due to the already mentioned failure in epipolar geometry.

The second error image shown in figure 4 uses equation 7 to correct the higher order distortion. Overall error is diminished but since nothing has been done to improve matching the top corners are still not corrected. For a clearer perception of the corrected distortion see figure 5 which shows the 3D reconstruction of the same plane (they are translated in relation to one another for visualization only) with (bottom plane) and without (top plane) use of equation 5. The plane

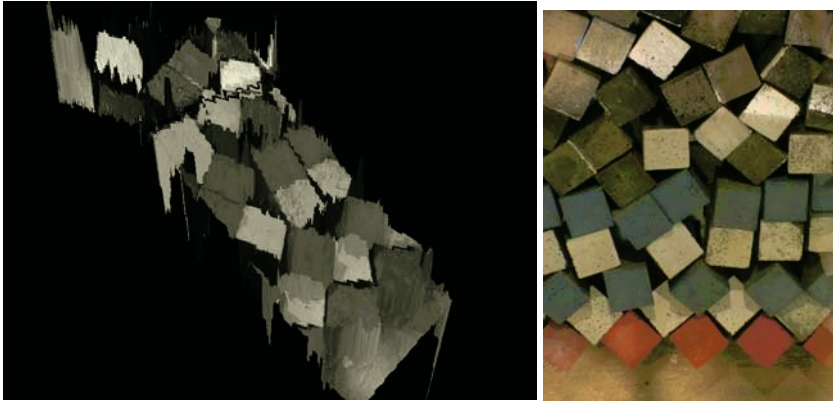




**Fig. 5.** 3D comparison of plane reconstruction with snell correction applied and without it.



**Fig. 6.** 3D view and left image of a model breakwater partially submerged.



**Fig. 7.** 3D view and left image of another model breakwater partially submerged.

reconstructed as a paraboloid effect mentioned earlier is clearly visible on the top plane. Although the planes are placed one above the other for comparison purposes, they are both at the same depth (1.5m).

Finally, the result of using bi-dimensional matching is shown in the third error image of figure 4. Note that only a few pixels (depending on the resolution, baseline and depth of the scene) need be searched away from the epipolar line, and only where the angle of incidence is greater than a certain tolerance. The maximum error is now 3 centimeters for the plane at  $z = -1.5\text{m}$ , which is the expected error due to the discrete nature of the sensor at the given distance.

### 3.2 Real World Experiment

Figures 6 and 7 show two reconstructions of a real breakwater physical model. The first uses images taken with video low resolution PAL cameras with a baseline slightly below 40cm and about 1.2m above the water. The second uses images taken with a beam splitter mounted on a 6 megapixel still camera. The baseline is about 5cm at 1.2m above the interface. Notice in both reconstructions the discontinuity near the top where the underwater and overwater reconstructions are fused. Unlike the synthetic images these are not so feature rich (for example dark shadows appear between rocks), resulting in some matching errors. Better results should be possible with algorithms that deal with occlusions and little texture.

## 4 Conclusion

We have shown how to diminish the refraction effect introduced by the presence of an interface between a stereo rig and the scene. The solution described allows for standard stereo matching algorithms to be used. The results show that the reconstruction error due to refraction is negligible, provided the cameras are looking perpendicularly to the water surface.

## References

1. G. Hough and D. Phelp. *Digital Imaging Processing Techniques for the Aerial Field Monitoring of Harbour Breakwaters*, 1998.
2. <http://www.vision.caltech.edu/bouguetj/>
3. Zhang, Z. *Flexible Camera Calibration By Viewing a Plane From Unknown Orientations*, Microsoft Research, 1999.
4. Heikkilä, J. and O. Silvin. *A Four-step Camera Calibration Procedure with Implicit Image Correction*, University of Oulu, 1997.
5. Pollefeys, M. *Tutorial on 3D Modeling from Images*, In conjunction with ECCV 2000, Dublin, Ireland, Jun, 2000.
6. Sun. C. “Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques”, *International Journal of Computer Vision*, vol.47 no.1/2/3, pp.99-117, Mai, 2002.
7. Demirdjian, D. and T. Darrell. “Using multiple-hypothesis disparity maps and image velocity for 3-D motion estimation”, Massachusetts Institute of Technology.

# A Functional Simplification of the BCS/FCS Image Segmentation\*

Pablo Martínez, Miguel Pinzolas, Juan López Coronado, and Daniel García

Universidad Politécnica de Cartagena, Dpto. Ingeniería de Sistemas y Automática  
Paseo Muralla del Mar s/n. 30202, Cartagena (Murcia), Spain  
{pablo.martinez,miguel.pinzolas,  
jl.coronado,daniel.garcia}@upct.es

**Abstract.** In this paper, a functional simplification of the BCS/FCS neurobiological model for image segmentation is presented. The inherent complexity of the BCS/FCS system is mainly due to the close modelling of the cortical mechanisms and to the high number of parameters involved. For functional applications, the proposed simplification retains both the biological concepts of the BCS/FCS and its performance, while greatly reducing the number of parameters and the execution time.

## 1 Introduction

Image Segmentation has been studied for decades by researchers of animal and computer vision. The present *state of the art* computer vision systems do not even approach the performance of human vision in image understanding, proving that there is still much to be learned from biological vision systems. With this in mind, many computer vision researchers have chosen biomorphic engineering approaches as the neural networks.

The BCS/FCS [1-4] is a neural network system for boundary segmentation and surface representation, inspired by a model of visual processing in the cerebral cortex. This model retains part of the biological concepts in which it is based.

Neural network interactions between two subsystems: BCS (Boundary Contour System) and FCS (Feature Contour System) are the basis of this model. These interactions are produced in the human visual cortex once the lateral genicular nucleus (LGN), which regulates flow from retina to primary cortex, preprocesses the image which gets “contrast enhanced”. BC system interacts FC system, complementing one to each other in order to delimit surfaces in the scene. As a result, invariant properties of surface shape are usually perceived with high fidelity, despite gross perturbations of surface appearance. The information about variable aspects of the objects is eliminated or treated as noise [3]. The Boundary Contour system (BCS) model detects and completes coherent edges that retain their sensitivity to image contrasts and locations, performing a perceptual grouping. The Feature Contour system (FCS) model compensates for local contrast variations and uses the compensated signals to diffusively fill-

---

\* This work has been supported in part by the Spanish Ministerio de Ciencia y Tecnología, under grant VICTOR: TIC2000-0406-P4-05.

in surface regions within the BCS boundaries, so that subsystem is responsible for brightness and surface perception.

Summarizing, BCS/FCS performs the enhancement and conditioning of images acquired by the visual system. This module is the result of the analysis of several and detailed experiments with the visual cortex of superior mammals as monkeys and humans, where the goal is to represent as close as possible the main aspects of neurobiological systems. However, the implementation of this system into the practical applications demanded by the industry is not easily feasible. Problems like the processing time (mainly due to the recursive nature of the processing), complexity in the tuning of the intervening parameters (the kind and condition of the scenes processed can vary greatly and these parameters have to be tuned accordingly), and the lack of optimal performance due to the limitations of the biological approach have to be avoided in some way.

In this work, a simplification of the forming stages is presented. All the concepts contained in the BCS/FCS are used in the new implementation but restraining the complexity and making it more functional. This is accomplished in two main ways: the first one is by the reduction in the number of parameters to tune (simplifying functions) which contributes to make this algorithm less dependant of the kind of images used for each application. The second way towards adapting this system for practical applications is to reduce the processing time by means of restraining recursivity and operations performed in each stage. By means of this action the processing speed is increased in a high rate and, although the model is not so close to biological aspects, the core of the system still retains the main concepts of the BCS/FCS.

## 2 Conceptual Description of the BCS/FCS Neural Network

The BCS/FCS neural network model was originally developed by Grossberg & Mingolla [1-3] through a detailed analysis of biological vision. This is a partial model of the human visual system and reveals how it detects, completes, and cleans from noise and useless information general boundaries. The segmentations produced are based in regions of different texture, color or luminance.

The lower level of the system (Stage 1) is a conditioning operation which boosts the contrast, normalizes the brightness in the input image and simultaneously reduces the speckle noise [5, 7]. It is performed by cells at the retinal and Lateral Genicular Nucleus. Receptive fields of these cells (see Figure 1), with an isotropic (not sensitive to orientation) center-surround structure, are the core of this stage. Two output channels, convolving the input image with a combination of two Gaussian functions of different size ( $\sigma$ ), are obtained: one of them detects transitions in the input image from dark to light - ON channel:

$$X_{ij}^{g+} = \left[ \frac{AD^+ + S_{ij}^c - S_{ij}^{sg}}{A + S_{ij}^c + S_{ij}^{sg}} \right]^+ \quad (1)$$

The other detects transitions from light to dark - OFF channel:

$$X_{ij}^{g-} = \left[ \frac{AD^- + S_{ij}^{sg} - S_{ij}^c}{A + S_{ij}^c + S_{ij}^{sg}} \right]^+ \quad (2)$$

$S_{ij}^c$  and  $S_{ij}^{sg}$  are the convolution of the input image with Gaussians of different width ( $\sigma$ ).  $A$ ,  $D^+$  and  $D^-$  are parameters depending of the nature of the input images. Along this work, the ‘+’ superscript means that only positive values are considered, while negative values are truncated to zero.

The combination of these channels produces the output of the Stage 1. These channels do not respond to uniform light in the input image.



**Fig. 1.** Left: Receptive field of an ON channel. Right: Receptive field of an OFF channel.

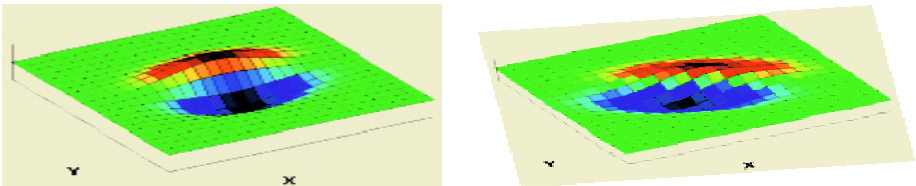
The outputs of these cells excite receptive fields at Stage 2. The function of this second module is mainly the segmentation of the existing borders. Also, a slight smoothing action on the surfaces enclosed by these borders is performed. It is formed by pairs of simple cells (which are directional) with the same orientation, which are sensitive to opposite contrast polarity. Their receptive fields, as can be seen in Figure 2, detect either an increase or a decrease of the activation in their preferred direction. The simple cell pairs, in turn, pool their rectified and oppositely polarized output signals at complex cells with the same orientation. These complex cells are not sensitive to direction of contrast. They respond equally well to increase/decrease of intensity. Conceptually, in this module difference of elongated Gaussians (rotated for processing several orientations) is convolved with the output image of the Stage 1. The global output image (for each scale  $g$ ) for this stage is obtained from the sum of the resulting processing for each orientation  $k$  (twelve in this work). The output of the simple cells is modeled by the equations:

$$s_{ijk}^{Rg} = \left[ (R_{ijk}^{g+} + L_{ijk}^{g-}) - (R_{ijk}^{g-} + L_{ijk}^{g+}) \right]^+ \tag{3}$$

$$s_{ijk}^{Lg} = \left[ - (R_{ijk}^{g+} + L_{ijk}^{g-}) + (R_{ijk}^{g-} + L_{ijk}^{g+}) \right]^+ \tag{4}$$

where  $R$  and  $L$  are the four convolutions of the ON and OFF channels from Stage 1 with the two elongated Gaussians. The output of the complex cell for each orientation ( $k$ ) and scale ( $g$ ) is:

$$c_{ijk}^g = s_{ijk}^{Lg} + s_{ijk}^{Rg} \tag{5}$$



**Fig. 2.** Left: horizontal simple cell. Right: diagonal simple cell.

Stage 3 is conformed by a cooperative-competitive loop. This recursive procedure enhances the segmentation process by the completion of the discontinued borders and broken connections in collinear segments belonging to the same border (cooperative action) and by destroying false parallel contours, reducing the noise, and attenuating the presence of perpendicular lines which could belong to smaller structures and objects without real interest for the segmentation (competitive action). The boundary completion is made by the bipolar cells, which act as logical AND functions for collinear borders (Figure 3). If both lobes of the cell coincide with collinear lines when convolving with the image then these lines will be joined. The competitive and cooperative modules interact one to each other.

The output of the boundary competition is:

$$Y_{ijk}^g = \left[ \frac{BE_{ij}^{3g} - CI_{ijk}^{3g}}{A + E_{ij}^{3g} + I_{ij}^{3g}} \right]^+ \quad (6)$$

where  $E_{ijk}^{3g}$  is the pondered combination of the output of Stage 2 and the cooperative module:

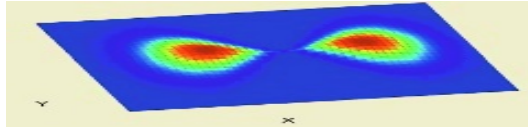
$$E_{ijk}^{3g} = G_f C_{ijk}^g + G_b Z_{ijk}^g \quad (7)$$

The output of the cooperative module is:

$$Z_{ijk}^g = \left[ \frac{BE_{ijk}^{4g}}{A + E_{ijk}^{4g}} \right] \quad (8)$$

where  $E_{ijk}^{4g}$  is a combination of the output of the competitive module and several convolutions of the bipole cells with  $Z_{ijk}^g$ .

$$E_{ijk}^{4g} = Y_{ijk}^g + H_{ijk}^{4g} \quad (9)$$



**Fig. 3.** Lobes of a horizontal bipole cell.

The three stages commented previously (LGN stage, simple and complex cells and cooperative-competitive loop) form the so-called Boundary Contour system (BCS).

Boundary Contour System establishes a barrier to the filling-in (Stage 4) of the surfaces delimited by the boundaries. The system that carries out this filling-in is the Feature Contour System (FCS). For image pixels through which no boundary signals pass, the resulting intensity values become more homogeneous as diffusion evolves; but when boundary signal intervene they inhibit the diffusion, leaving a resulting activity difference on either side of the boundary signal [5]. This diffusiveness operation is an iterative task intervened not only by boundaries obtained from BCS but also by ON and OFF cells from Stage 1. Figure 4 shows an example of diffusion.

The diffusive filling-in obeys the equations:

$$F_{ij}^{g+} = \frac{X_{ij}^{g+} + \sum_{p,q} F_{pq}^{g+} P_{pqij}^g}{D + \sum_{p,q} P_{pqij}^g} \tag{10}$$

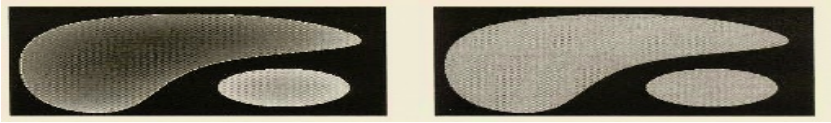
$$F_{ij}^{g-} = \frac{X_{ij}^{g-} + \sum_{p,q} F_{pq}^{g-} P_{pqij}^g}{D + \sum_{p,q} P_{pqij}^g} \tag{11}$$

The boundary-gated permeabilities obey

$$P_{pqij}^g = \frac{\delta}{1 + \varepsilon(y_{pq}^g + y_{ij}^g)} \tag{12}$$

where

$$y_{ij}^g = \sum_k Y_{ijk}^g \tag{13}$$



**Fig. 4.** Brightness diffusion in FCS. Left: two objects presenting brightness differences due to shading. Right: after the FCS processing, the two objects now present uniform brightness.

The last stage (Stage 5) is the scale averaging or combination of scales. The final output image is attained by a weighted combination of the resulting images at different scales. The weight for each scale in the global result is heuristically evaluated, depending on the nature of the images to process.

### 3 Simplification of the Original Algorithm

Due to the need of getting a higher processing speed and the goal of eliminating as many parameters as possible, several changes have been made to the last development of the BCS/FCS by Grossberg et al. [7].

Stage 1 has not been changed. The difference of Gaussians (similar to the Laplacian of Gaussian function) is the best biological approach to date for the modelling of the LGN cells. Although there are several filters similar in performance to the Difference of Gaussians function, this filter combines the edge enhancement property with the removal of the high frequency noise retaining the biological approach.

Stage 2 is composed of directionally sensitive receptive fields (simple cells) which detect increase or decrease of activation in their preferred direction. The complex cells, also included in this stage, pool the information from simple cells obtaining the borders of the image. The receptive field of these cells is similar to the receptive field of the ON channels and OFF channels. The difference between them causes the anisotropy. This elongation is the essence of the border detection. In the simplified algorithm, this essence has been preserved but implemented in a simpler way. For example, for the detection of an edge, in Grossberg’s original system two elongated difference of Gaussians (simple cells) are applied in the same area for detection of increase and decrease of activation respectively for a given direction (twelve, in this work), so the outputs of these two simple cells (one for detecting increase of contrast

and the other for contrast decrease) are pooled. In the system presented in this work, only one simple Sobel approximation for the gradient (rotated for each orientation) is convolved for a given orientation, which reduces the computational load more than a half for this module. In the original system, the convolutions are made using the two elongated Gaussians with the ON and OFF channels, so four different outputs are obtained. An arithmetic combination of these four resulting images is used for the global output of this stage. We have found that this is functionally equivalent to convolving the combination of the ON and OFF channels with a rotated Sobel operator for each orientation. Therefore, the number of necessary convolutions is reduced in a quarter and the results obtained are on a par without losing the main concept involved.

In Stage 3, some modifications have been arranged in order to speed up the processing. This module is divided into two sub-systems, the first one being the Competitive Stage. It performs a useful cleaning of false and residual borders in the images, by means of an iterative process in which Stage 2 and the Cooperative Stage are involved. This process is computationally very expensive, so its implementation has been changed. In spite of implementing a separate task interacting with the cooperative action, the two modules have been joined in the algorithm. This integration obeys the equation:

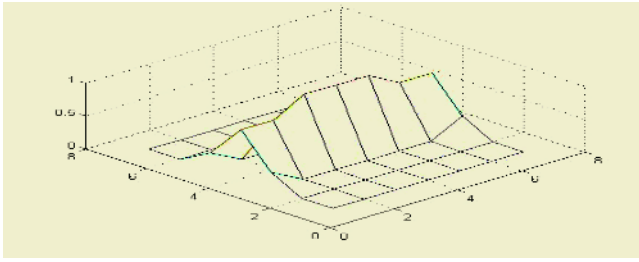
$$Y_{ijk}^s = \sum_k c_{ijk}^s \cdot \cos \left[ 2 \cdot (K_1 - K) \cdot \frac{\pi}{N_{or}} \right] \cdot W_{ijK_1} \quad (14)$$

where  $|K_1 - K| < 2$  or  $\left| K_1 - K - \frac{N_{or}}{2} \right| < 2$ , being  $N_{or}$  the total number of orientations considered,  $K$  the analyzed orientation for the image ( $0 \leq K \leq N_{or}$ ) and  $K_1$  the orientation of the  $W$  filter ( $0 \leq K_1 \leq N_{or}$ ). This filter is a particularization of the general dipole receptive field given in [7], and it is shown for  $0^\circ$  orientation in figure 5.

Only the three closest orientations to the perpendicular to the border are considered for the competition. This competition attenuates all the borders not belonging to the real contour of the blob. In the original system all the orientations have been considered as shown in equation (6). This is a more correct approximation but also slower to process while leading to similar results.

The second subsystem in the CC loop is the Cooperative. The function defined for the bipole cells has several adjustable parameters related to the response saturation level, the threshold for the firing of the lobes, the length of both lobes from the filter center, the spatial deviation from co-linearity and the orientational deviation from co-linearity. All of these adjustable values broaden the field of utilization for this system allowing fine-tuning the parameters according to the nature of the input images. A much simpler function has been chosen for replacing the original bipolar cells as shown in figure 5. This alternative has two great advantages. First, the filter size is considerably reduced. Considering that this stage has recursive implementation and that this mask must be convolved at each iteration with all the orientations, reduction in the size of the filter has a very evident effect in the efficiency. The second advantage is the elimination of the need to tune the commented parameters, which is an arduous task.





**Fig. 5.** Filter ( $W$ ) replacing the original receptive field of a bipole cell.

Stage 4 is the Surface Filling-in process. As for Stage 1, no major changes have been made with respect to the original system. The interaction with Stage 1 has been limited to the output image from that stage and not performed with the ON and OFF channels separately, so the processing time is reduced on a half.

Stage 5 (scale averaging) has been removed from the algorithm. After several experiments, the conclusion was that no perceptible enhancement over the results obtained having in account only one scale was obtained. This is due to the previous fine-tuning of the intervening parameters dependant of the size of the blobs of interest. The smaller and the greater scales are not processed. Considering the three scales, the processing time would have been multiplied by three, a major inconvenient for the purpose of this system.

## 4 Results

A set of basic images used for the analysis and comparison of the results obtained with the original neural model and the modified one are presented in Fig. 6. These images have been selected because they allow to test and validate the performance of the most interesting stages comprising the BCS/FCS. For example, in the leftmost image the broken boundaries in the polygon allow to evaluate the completion of the boundaries in several directions (Stage 3). Also, the residual blobs existing in this image are reduced in part by the Stage 1 and by the competition sub-stage in Stage 3. In all the images, the noise (a kind of salt and pepper) is eliminated by the difference of Gaussians in Stage 1. The center and right images present illusory contours. These illusory contours can be recognized without actually being seen, and are easily perceived by humans, but hardly detected by computer vision systems. This is due to the existence of underlying textures of parallel lines. The ends of these lines form a line of disjointed points which excite the bipole cells of the cooperative module (Stage 3). In the three images, surface diffusion has been carried out by the FCS system, and should lead to the differentiation of several uniform surfaces, corresponding to zones of the original image delimited by borders (real or illusory), or that present a common texture (like the different line densities in the center image of Fig. 6).

The obtained images processed using both BCS systems are presented in Table 1. In Table 2 the results of the FCS for the simplified method are shown. No changes from original system have been made in this stage.

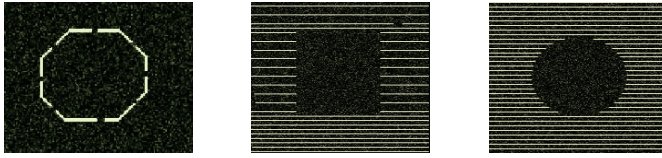


Fig. 6. Three test images.

Table 1. Images obtained after BCS processing.

Original BCS			
Simplified BCS			

Table 2. Images obtained after FCS processing.

--	--	--

As can be seen in the results presented in the left column of Table 1, the completion of the boundaries is successfully performed. The variance in the thickness of the border and the difference in brightness between the two images (up and bottom) are due to the different functions convolved with the input images in the cooperative stage. The two columns on the right show the similarity in the results for the completion of illusory contours with both methods.

In table 2 the intensity values become more homogeneous as diffusion evolves. In the left figure the borders of the object inhibit the diffusion outside the boundaries. The same process of diffusion has been initiated outside the object, but at a lower level. In the centre and right images, the diffusive process continues the work initiated by the BCS system. Several uniform regions are finally obtained, corresponding to those zones inside real or illusory borders, or distinct textures, as expected.

## 5 Discussion

A new functional and simplified implementation of the Neural BCS/FCS model for image processing is presented. While retaining the functionality and the biological inspiration of the original algorithm, this simplification eases its use and considerably

speeds-up the processing. As can be seen in the experiments included in this work, when applying this simplified revision of the neural system two goals are achieved: the processing time is reduced considerably and the tuning of the algorithm is made easier avoiding the need to adjust by hand a great amount of parameters, which introduces an arduous and time demanding task to the system programmer.

## References

1. Grossberg S., Mingolla E., (1985). Neural Dynamics of Form Perception: Boundary Completion, Illusory Figures, and Neon Color Spreading. *Psychological Review*, 92, 173-211.
2. Grossberg S., Mingolla E., (1985). Neural Dynamics of Perceptual Groupings: Textures, Boundaries, and Emergent Features. *Perception and Psychophysics*, 38, 141-171.
3. Grossberg S., Mingolla E., (1987). Neural Dynamics of Surface Perception: Boundary Webs, Illuminants, and Shape-from-Shading. *Computer Vision, Graphics, and Image Processing*, Vol. 37, 116-165.
4. Grossberg S., Todorovic, (1988). Neural Dynamics of 1-D and 2-D Brightness Perception: A Unified Model of Classical and Recent Phenomena. *Perception and Psychophysics*, 43, 241-277.
5. Grossberg S., Mingolla E., Williamson J. (1995). Synthetic Aperture Radar Processing by a Multiple Scale Neural System for Boundary and Surface Representation. *Neural Networks*. On special issue *Automatic Target Recognition*.
6. Levine M.W. and Shefner, J.M. (1991). *Fundamentals of sensation and perception*, 2nd ed. Pacific Grove, CA: Brooks/Cole.
7. Mingolla E., Ross W., Grossberg S. (1999). A Neural Network for Enhancing Boundaries and Surfaces in Synthetic Aperture Radar Images. *Neural Networks*, 12. 499-511.

# From Moving Edges to Moving Regions

Loic Biancardini<sup>1</sup>, Eva Dokladalova<sup>1</sup>, Serge Beucher<sup>2</sup>, and Laurent Letellier<sup>1</sup>

<sup>1</sup> CEA LIST, Image and Embedded Computer Laboratory  
91191 Gif/Yvette Cedex, France

{biancardini,eva.dokladalova,letellier}@cea.fr

<sup>2</sup> Centre de Morphologie Mathématique ENSMP  
35 rue Saint Honoré 77305 Fontainebleau cedex, France  
beucher@cmm.ensmp.fr

**Abstract.** In this paper, we propose a new method to extract moving objects from a video stream without any motion estimation. The objective is to obtain a method robust to noise, large motions and ghost phenomena. Our approach consists in a frame differencing strategy combined with a hierarchical segmentation approach. First, we propose to extract moving edges with a new robust difference scheme, based on the spatial gradient. In the second stage, the moving regions are extracted from previously detected moving edges by using a hierarchical segmentation. The obtained moving objects description is represented as an adjacency graph. The method is validated on real sequences in the context of video-surveillance, assuming a static camera hypothesis.

## 1 Introduction

Automated video surveillance applications have recently emerged as an important research topic in the vision community. In this context, the monitoring system requirement is to recognize interesting behaviors and scenarios. However, in such a system, the main problem is to localize objects of interest in the scene. In this context, every moving area is potentially a good region of interest.

There are three conventional approaches to automated moving target detection: background subtraction [5–7, 13], optical flow [5, 8] and temporal frame differencing [5, 10, 14]. In video surveillance, the background subtraction is the most commonly used technique. However it is extremely sensitive to dynamic change of lighting. Nevertheless, it requires a prior knowledge of the background, which is not always available. In the second category of methods, the optical flow estimation is used as a basis for further detection of moving objects. However, it is a time consuming task. It is affected by large displacements and does not provide the accurate values, neither at moving objects contours, nor in large homogeneous areas.

In this paper, we focus on the temporal frame differencing methods. These techniques enable fast strategies to recover moving objects. However, they generally fail to extract accurately both slow and fast moving objects at the same time. In such case, a tradeoff between missed targets and false detections is very

hard to obtain. To overcome these problems, we first propose a new difference scheme suited to moving objects boundaries detection. Then, a hierarchical segmentation [1–3] of the current frame is used to complete these contours and extract the underlying moving regions.

The paper is organized as follows: section 2 introduces the method for motion boundaries extraction. In section 3, the use of the hierarchical segmentation to retrieve the moving regions is described. The experimental results are presented in section 4. Then, we give the conclusions on the proposed method and we discuss the future work.

## 2 Moving Edges Detection

The frame differencing methods take advantage of occlusions, which occur at moving objects boundaries. Various kinds of approaches have been attempted in the literature [10, 11, 13, 14]. Generally, the presence of the occlusions is detected using the absolute difference of two successive frames. However, the occlusions do not correspond to the position of the true object boundaries neither in the first image nor in the second one. Moreover, depending on frame rate and speed of the moving objects, the difference map can critically differ. When an object moves slowly, image intensities do not change significantly in its interior. Consequently, the resulting difference image exhibits high values only at motion boundaries. In the opposite case, if the object has completely moved from its position, the resulting frame difference will exhibit high values inside the object body in both images. It is the so-called ghost phenomena [12] and leads to false detections.

In [13, 14], the authors propose to use a double-difference operator. The frame difference is performed on the two pairs of successive images at time  $(t-1, t)$  and  $(t, t+1)$ . Then, the result is obtained by the intersection of these two difference maps. However, when an object moves slowly this intersection may be reduced to an insufficient number of pixels.

### 2.1 Difference Scheme

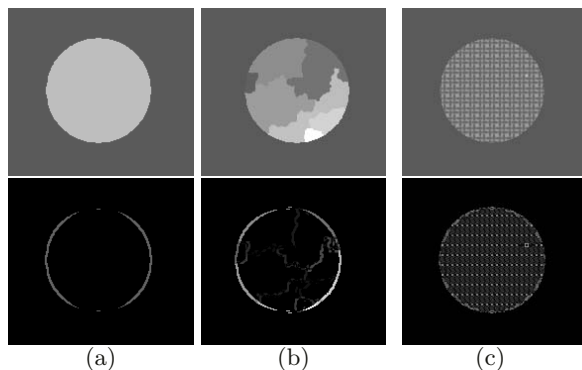
In the following,  $I^t : \mathbb{Z}^2 \rightarrow \mathbb{N}$  indicates a discrete image at a given time  $t \in (0, T]$ . We note the reference frame, the frame in which we want to localize and segment the objects in motion.

The proposed method considers three successive images  $I^{t-1}$ ,  $I^t$  and  $I^{t+1}$ . We assume that moving edges position depends rather on the gradient changes in the successive images than in the images themselves.

First, we compute the spatial gradient modulus of  $I^{t-1}$ ,  $I^t$  and  $I^{t+1}$  and we note  $g^t = \|\nabla I^t\|$  (respectively  $g^{t\pm 1}$ ). Then, the symmetrical frame difference is obtained on the two pairs of gradient images. The moving edges measurement at a given time  $t$  is defined as the infimum operator of the two difference maps:

$$mem^t = \inf(|g^{t+1} - g^t|, |g^t - g^{t-1}|) \quad (1)$$

The infimum operator properties and the analysis of gradient over three frames yield the interesting behaviors:



**Fig. 1.** Results of *mem* on three different cases: (a) homogeneous region, (b) assembly of homogeneous regions and (c) textured area

1. a maximum response at moving objects boundaries locations: when a contrasted object is moving over a homogeneous area, the *mem* is equal to the original gradient in the reference frame.
2. a significant robustness to motion amplitude: in the case of fast moving objects, the result is not delocalized and the ghost phenomena are drastically reduced.
3. a significant robustness to random noise (non-repeatable in subsequent frames).

However, due to low motion, weak contrast with the scene and the aperture problem (sliding contours), the moving edges measurement will certainly fail to provide information along the whole contours of a moving object (figure 1). The forthcoming section explains how to overcome this problem in order to obtain reliable moving regions.

### 3 From Moving Edges to Moving Regions

In this section, we propose a method to extract moving regions based on the new moving edges measurement (*mem*) proposed in paragraph 2.1. However, the *mem* operator does not result in the complete object contours. Thus we propose to consider an additional information issued from a spatial segmentation of the reference image. Nevertheless the segmentation process generally results in an over segmentation of the image, an accurate description of the image requires multiple levels of details. Thus, in our approach, the moving regions are searched through the levels of a hierarchical segmentation, which allows to study the regions at different scales.

We start by extracting an initial set of moving contours corresponding to spatial edges with a sufficient *mem* value. Then, the moving objects are detected by browsing a set of candidate regions extracted from a hierarchical partition.

### 3.1 Hierarchical Segmentation and Candidate Set to Detection

Some attempts to extract the meaningful image regions by gathering the regions of an initial segmentation can be found in [1, 2, 4]. However, they are not based on exhaustive analysis of region grouping which have a significant computational complexity. As explained in these publications one way to reduce the number of candidates is to build a hierarchical segmentation. After an initial partition is built, a graph is defined, by creating a node for each region and an edge for each adjacent regions pair. Graph's edges are weighted according to a dissimilarity criterion (as example, a grey level difference) between two regions. The hierarchical segmentation is obtained by progressively merging regions of the initial segmentation, in an increasing dissimilarity order. The process is iteratively repeated until only one region remains. By keeping track of the merging process, we construct the candidate set of regions  $C$ . Each time two regions merge, the resulting region is added to the candidate list. Note that the candidates are sorted according to their level of apparition in the hierarchy. The total amount of distinct regions in the candidate list is  $2N-1$  ( $N$  is the number of regions of the initial partition) [1]. This hierarchical segmentation only contains the more meaningful assembly of regions in the sense of the chosen dissimilarity criterion.

In our approach, we use the set of contours and regions given by the watershed transform proposed in [3]. We choose a robust dissimilarity criterion based on the contrast: for a given regions pair, the value of the criterion is defined by the median value of image gradient modulus along the watershed lines separating the regions.

### 3.2 Initialization Step: Extraction of Moving Contours

Once the hierarchical segmentation is built, the next step of the algorithm is to extract a set of moving contours: the  $mem$  is calculated and a threshold is applied to obtain a binary image. A set of moving points designed as the most significant contours in motion ( $msem$ ) is obtained by intersecting the thresholded  $mem$  image with the lowest level's contours in the hierarchy. The resulting binary image (section 4, figure 3(b)) may not contain the whole moving object's boundary but only some incomplete and fragmented parts. Consequently, the next step of our method is to gather and complete moving contours coming from a same object, and discard small or isolated components corresponding to residual noise. True moving edges are supposed to be distributed with enough coherence and density around a same region to be gathered as the contours of this region. *A contrario*, noise components are sparse and dispersed. They can not be assembled as the contours of any region in the hierarchy.

### 3.3 Detection of Moving Regions

The detection step is achieved by independently optimizing a local criterion on each region of the candidates list defined in section 3.1. In the following, for any given candidate  $C_i \in C$ , the frontier  $\partial C_i$  of the region is defined as the subset of

watershed points enclosing  $C_i$ . The matching score of a region  $C_i$  is calculated as the proportion of significative contours in motion contained in its frontier  $\partial C_i$ . This is simply expressed by:

$$ms(C_i) = \text{card}(\partial C_i \cap mscm) / \text{card}(\partial C_i) \quad (2)$$

where  $\text{card}$  refers to the cardinal operator.

Each candidate is successively tested according to its order of apparition in  $C$ . A candidate is labeled as detected if its score  $ms(C_i)$  is higher than a predefined threshold  $T_{percent} \in [0, 1]$ .

The method may lead to some incorrect detections as depicted in figure 2(c). In figure 2(c), the region  $C_2$  which causes the error is detected because its frontier in common with the moving region  $C_1$  (figure 2(b)) is quite long. Nevertheless, the frontier of  $C_2$  is longer than the frontier of  $C_1$  but does not contain more moving contours points (figure 2(a)). Consequently, the score  $ms(C_i)$  of the region  $C_1$  is higher than the one of  $C_2$  which enables to select the final correct region.

As explained in section 3.1, each candidate (except those from the initial partition) comes from a merging sequence of some preceding candidates. Owing to the construction of  $C$ , when  $C_i$  is tested as a moving region, it implies that all the grouping candidates constituting this sequence are already processed and their scores are known. In the following, we will refer to this set of candidates as the set of Ancestors of  $C_i$ . To avoid situations such as depicted in figure 2, we propose to add the following condition to the detection: If  $C_i$  exhibits a score superior to  $T_{percent}$ ,  $C_i$  is said to be detected if and only if, its score is higher than any score of its ancestor. This can be expressed by adding the following condition:

$$ms(C_i) > \max_{C_k \in \text{Ancestors}(C_i)} (ms(C_k)) \quad (3)$$

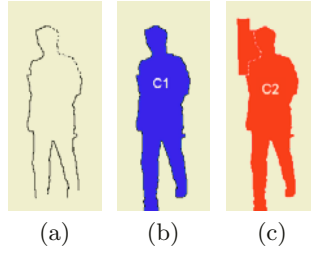
This is a sufficient way to discard many false detections. However, when candidate regions correspond to higher levels of the hierarchy, their frontiers are longer. Thus, a significant score is more difficult to obtain (a larger portion of the contour may be missing). In that case, the score can not be constrained to be strictly higher than the previous ones. Consequently, we propose to test whether the new matching score is higher than the previous ones multiplied by a weighting coefficient  $\alpha$ , taken in the interval  $[0,1]$ .

## 4 Experimental Results

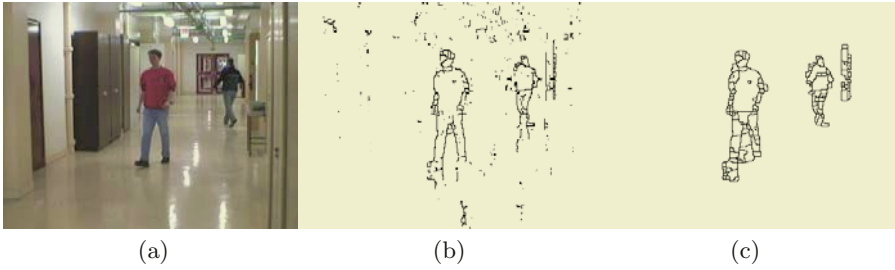
In this section, we present some results on video sequences corresponding to real situations of video-surveillance.

In the presented results, we use the regularized Deriche gradient [9] to obtain the moving edges measurement (see section 2). The regularization parameter  $\sigma$  should be chosen greater than 2.0 in order to preserve poorly contrasted or narrowed structures.





**Fig. 2.** (a) the (*mscm*) set (b) a first candidate matching the contours (c) a later candidate matching the contours



**Fig. 3.** (a) original image, (b) set of moving contours initially detected (c) contours of detected regions after parsing the hierarchy

The threshold parameter  $T_{mem}$  used to obtain the *mscm* mainly depends on the level of noise in the *mem* image. Nevertheless the experiments show that it is stable over time for a given scene and a fixed video camera. In all the presented experiments this parameters is set  $T_{mem} = 2.0$ . The result of *mscm* detection is presented in figure 3(b).

As it was presented in the section 3, we use the watershed transform to obtain the initial segmentation. In order to reduce (once again) the computational cost of the algorithm we propose to use a reduced set of markers. It is obtained by the h-minima operator with  $h = 3$  [15]. During the detection process, we use  $T_{percent}$  to express the ratio of the target boundary length, which can be missing without altering the detection of the corresponding region. This parameter was set to 0.65, which enables to detect the regions from an incomplete set of moving contours, without generating false detections. The experimentally verified best values range of parameter  $\alpha$  is [0.65, 0.85]. The *alpha* paramater's influence can be reduced by taking into account the size ratio of the currently treated region and its detected ancestors in the algorithm of section 3.3.

The initial set of moving edges is presented in the figure 3(b). The contours of all the regions detected during the matching process are shown in figure 3(c). Once the detection is achieved, isolated components with area under 50 pixels are removed and the remaining regions are merged according to the dissimilarity criterion. The results of this post-processing step are shown in figures 4(a) and 4(b).



**Fig. 4.** From top-left to bottom-right (for each data set): first image of the sequence and moving regions detected in some subsequent frames

## 5 Conclusions

This paper focuses on the extraction of moving objects in the video-surveillance context. The goal is to detect all potential zones of interest and create their representation suitable for tracking and scene interpretation.

First, we introduce a new method to perform the detection of moving objects boundaries. The moving edges are extracted with an operator based on the double differences of three successive gradient images. The defined operator is robust to random noise and the results are not affected by the displacement speed

of objects. Then we show how to use the hierarchical segmentation in order to pass efficiently from the incomplete detected contours to the entire regions in motion. To obtain the accurate set of moving regions, we propose to combine two criteria during the detection process: i) the contrast criterion ii) the matching score criterion. The hierarchical approach also reduces the computation time that is the limiting factor in the video-surveillance applications.

Another advantage of the method is that the extraction of the moving objects requires neither motion calculation nor prior knowledge of the scene.

In addition, the moving targets are extracted as an assembly of multiple homogeneous parts of different size and contrast. Due to the underlying hierarchical segmentation structure, their adjacency and inclusion relations are known. These considerations are very useful to construct a model for the detected targets. This model can be then used in several ulterior steps such as tracking, occlusions analysis or pattern recognition.

Consequently, the next stage of our work will concentrate on the study of the hierarchical graph-based object description for the scene interpretation and the object tracking in the security domain.

## References

1. P. Salembier, L. Garrido. Binary Partition Tree as an Efficient Representation for Image Processing, Segmentation, and Information Retrieval, .IEEE Transactions on Image Processing, 9(4):561-576, April 2000.
2. F. Zanoguera, B. Marcotegui, F. Meyer, A Toolbox for Interactive Segmentation Based on Nested Partitions, ICIP, Kobe (Japan), 1999
3. Beucher, Segmentation d'images et morphologie mathématique, Doctorate thesis, Ecole des Mines de Paris, Cahiers du centre de Morphologie Mathématique, Fascicule n° 10, Juin 1990.
4. Stan Sclaroff, Lifeng Liu: Deformable Shape Detection and Description via Model-Based Region Grouping. IEEE Trans. Pattern Anal. Mach. Intell. 23(5): 475-489 (2001)
5. D. S. Zhang, G. Lu, Segmentation of Moving Objects in Image Sequence: A Review, Circuits, Systems and Signal Processing (Special Issue on Multimedia Communication Services), 20(2):143-183, 2001.
6. Srinivas Andra, Omar Al-Kofahi, Richard J. Radke, and Badrinath Roysam, Image Change Detection Algorithm: A systematic Survey, Submitted to IEEE Transactions on Image Processing, July 2003.
7. M. Piccardi, Background subtraction techniques: a review, in Proc. of IEEE SMC 2004 International Conference on Systems, Man and Cybernetics, The Hague, The Netherlands, October 2004.
8. M.J. Black, P. Anandan, The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields, Computer Vision and Image Understanding, CVIU, 63(1), pp. 75-104, Jan. 1996.
9. R. Deriche, Fast algorithms for low-level vision, IEEE-PAMI, 12(1):78-87, 1990.
10. N. Paragios, R. Deriche, A PDE-based Level Set Approach for Detection and Tracking of Moving Objects, In Proceedings of the 6th International Conference on Computer Vision, Bombay,India, Jan. 1998.

11. Shi, Malik, Motion segmentation and tracking using normalized cuts, University of California, Berkeley report n°UCB/CSD-97-962, 1997
12. R. Cucchiara, C. Grana, M. Piccardi, A. Prati, "Detecting Moving Objects, Ghosts and Shadows in Video Streams" in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, n. 10, pp. 1337-1342, 2003
13. K. Toyarea, J. Krumm, B. Brumitt, and B. Meyers, Wallflower: Principles and practice of background maintenance," in International Conference on Computer Vision, 1999, pp. 255-261.
14. Yoshinari, K. and Michihito, M., "A human motion estimation method us using 3-successive video frames Proc. of Intl. Conf. on Virtual Systems and Multimedia, 1996, pp. 135-140.
15. Tae Hyeon Kim, Young Shik Moon, "A New Flat Zone Filtering Using Morphological Reconstruction Based on the Size and Contrast," VLBV, 1999

# Polygon Optimisation for the Modelling of Planar Range Data

Samuel Nunes, Daniel Almeida, Eddy Loke, and Hans du Buf

Vision Laboratory, University of Algarve, 8000-810 Faro, Portugal

Tel: +351 289 800900x7761, Fax: +351 289 818560

{loke,dubuf}@ualg.pt

<http://w3.ualg.pt/~dubuf/vision.html>

**Abstract.** In this paper we present efficient and fast algorithms for the reconstruction of scenes or objects using range image data. Assuming that a good segmentation is available, we concentrate on the polygonisation, triangulation and optimisation, i.e. both triangle reduction and adaptive edge filtering to improve edge linearity. In the processing, special attention is given to complex edge junctions. In a last step, vertex neighbourhoods are analysed in order to robustly attribute depth to the triangle list from the noisy range data.

## 1 Introduction

Range images obtained by laser cameras or other devices allow to construct a 3D model of an object or a scene. Normally, the processing required consists of (a) range data segmentation and (b) the modelling (triangulation) of the segmented data and the attribution of depth to the vertices of the triangle list. A lot of effort has been devoted to segmentation, see e.g. [1, 2] for a quantitative comparison of existing solutions. Recently, we extended our single-feature segmentation algorithm in a quadtree to multi-feature, i.e. instead of using only one component of normal vectors, computed by considering three adjacent pixels that form a triangle, we use the three components [3]. Here we will focus on the 3D reconstruction and not on the segmentation.

This paper concentrates on the second processing step, i.e. the modelling and the attribution of depth, using methods that are extremely simple and fast. Processing speed becomes important when many views of a rotated object need to be integrated for constructing a consistent and complete 3D model. Another, complex and slower modelling approach is the one of Khalifa [4], who uses bilinear Bézier patches for planar regions and NURBS patches for spherical regions to construct surface CAD models. Figure 1 shows the main steps in our processing pipeline, which will be explained in subsequent sections.

## 2 Filtering and Edge Detection

In a first pass, a range image is filtered in order to correct isolated pixels with values that differ completely from their neighbouring pixels. This filtering is

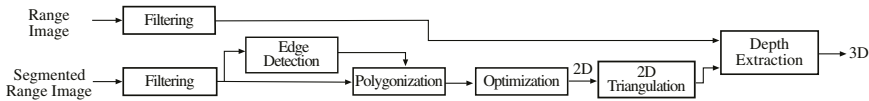


Fig. 1. Processing steps excluding surface shading and texture mapping.

important because it highly improves the depth extraction (Section 5). Using a neighbourhood size of  $3 \times 3$ , the centre pixel is compared with its 8 neighbours. When the difference does not exceed a threshold, the value of which may depend on the noise of the data, the neighbour is counted as being similar. If the total count is below 2, the centre pixel is assumed to be an outlier. Its value is replaced by the average of the biggest group of pixels with similar values, after analysing all combinations of the 9 pixels in the neighbourhood. We note that this filter gives much better results than a simple median filter, because it corrects small regions with different values that are often found at long edges of range images. The same type of filtering is applied to the segmented range image, but serves to correct single-pixel regions. In this case, an isolated pixel is substituted by the value of the majority of its neighbours.

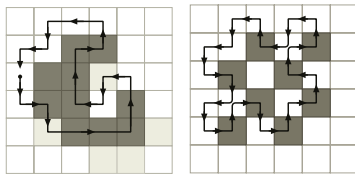
In the second step, edges are detected in the segmented range image. Here, edges (or transitions) are considered as geometric primitives on the discrete lattice that encode all available shape information at the pixel level. The result is another image, initialised with zero, that contains ones in the form of continuous and closed contours. This discrete representation can be obtained by applying a very simple operator: if the values of the 4 bottom-right pixels in a  $3 \times 3$  neighbourhood are not equal, the centre pixel is marked as an edge. Below, we use “edge” to refer to edge contours and “edge pixel” when addressing single edge points.

### 3 Polygonisation

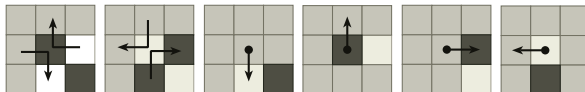
The edges detected in the segmented range image can now be used to create discrete polygons of all segmented regions (detected planar object faces). This is done by tracking the edge of each region, as shown in Fig. 2. At each position, the direction of the next pixel is determined from the  $3 \times 3$  neighbourhood, as shown in Figure 3. The first 2 cases to the left are tested first, and these depend on the previous path. The other 4 cases are only tested if the first 2 don’t match. We note that, if the segmentation contains a chessboard pattern consisting of single-pixel fields as shown in Fig. 2 (right), only the *outline* of the pattern will be tracked. However, such a pattern is impossible because of the filtering referred to above.

### 4 Mesh Optimisation

Up to this point we have discrete polygons that consist of many edge pixels. These edge pixels are vertices, but only in extremely simple cases, for example



**Fig. 2.** Examples of edge tracking in different regions.



**Fig. 3.** Edge-tracking rules: the position of the next pixel is determined from the  $3 \times 3$  neighbourhood. Black pixels belong to the region, white ones don't. Grey pixels are don't cares.

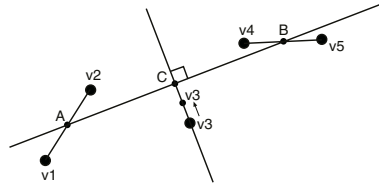
a straight horizontal or vertical edge, vertices can be eliminated. Optimisation aims at reducing the number of vertices of the discrete polygons, while preserving the geometry, and consists of the following steps: (a) iterative vertex filtering, (b) path extrapolation at bifurcating vertices, (c) vertex reduction, and (d) triangulation.

Iterative filtering must be applied to all distinct parts that make up the polygons, processing only once the parts that are shared between two neighbouring polygons. The shared parts are often isolated by special vertices with more complex junctions, like Y, T and K junctions. Such vertices we simply call “forks” because of the bifurcations, and the edge between two forks is called “path.” Every path starts and ends at a fork.

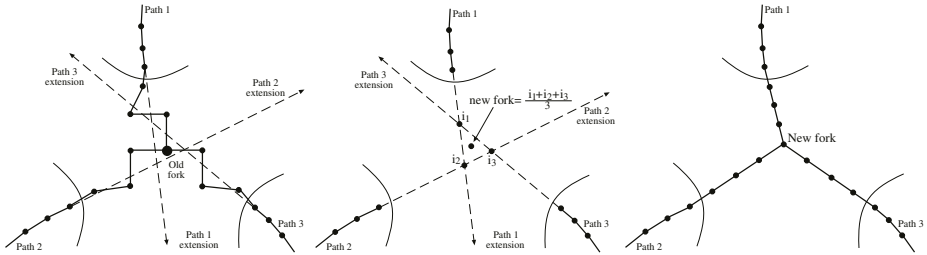
**Iterative Filtering:** This adaptive filtering is done in floating point, but we keep working with the discrete vertex lists of all paths between forks. The filtering is done by taking into account 5 successive vertices, moving only the position of the centre vertex. The positions of the first two vertices are averaged, as are those of the last two vertices. The straight line between the averaged two positions is used to move the centre vertex: it is moved perpendicularly towards the line such that its projected distance is halved, see Fig. 4. This is done iteratively for all vertices of a path. The absolute values of all movements are added, and the filtering of a path stops when the sum of a new iteration is below a certain threshold value. This filtering is more robust to noise than applying a mean or median filter.

**Path Extrapolation at Forks:** Iterative filtering will remove most noise from the paths. However, since the first 2 and the last 2 vertices of each path are not filtered, and because the fork vertices themselves are also affected by noise, special processing of forks and their neighbouring vertices is required.

The first step is to estimate the most probable position of a fork vertex, taking into account all paths that converge at the fork. For this, we assume that each path is approximately linear near the fork, and use vertices 4 and 5 from the



**Fig. 4.** Adaptive filtering of edges by considering two pairs of vertices, moving the centre vertex in the direction of the connecting line.



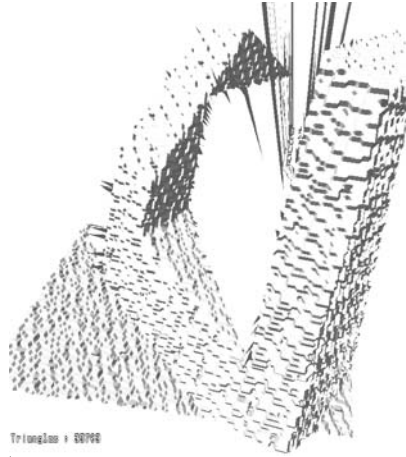
**Fig. 5.** Vertex correction by path extrapolation (left), intersection averaging (middle) and repositioning (right).

path to define such a line (the first 3 vertices are ignored because vertices 1 and 2 are not filtered and vertex 3 still may have a considerable error). An example of this is shown in Fig. 5 (left). The new coordinates of a fork will be the average of all intersections of the extrapolated lines (filtered paths), as shown in Fig. 5 (middle). However, if two lines are almost parallel, their intersection will not be used because of the possibly large error. If there are no useful intersections, the position of a fork will not be changed. Finally, given the new fork coordinates, the first 3 vertices of each path are interpolated between vertex number 4 and the new fork, see Fig. 5 (right).

**Vertex Reduction:** In order to reduce the number of vertices, we consider groups of 3 neighbouring vertices along each path, excluding fork vertices, and compute the angle between the first and second pair. The centre vertex is eliminated from the vertex list if the angle is close to 180 degrees, using a threshold value. After eliminating a vertex, we skip one vertex in order to avoid eliminating successive vertices in the same iteration, preserving the shape of a path. This process stops when zero vertices have been eliminated after an iteration.

**Triangulation:** The triangulation algorithm implemented is a very simple and straightforward one, see Chapter 1 in [5]. This method has a complexity of  $O(n^3)$ , with  $n$  being the number of vertices. For a better performance, faster but also more complex methods can be implemented, e.g. the Chazelle triangulation of  $O(n)$ . However, the main drawback of these algorithms is that they apply to simple polygons, and not to polygons with holes.





**Fig. 6.** Direct triangulation of part of a noisy range image.

## 5 Depth Extraction

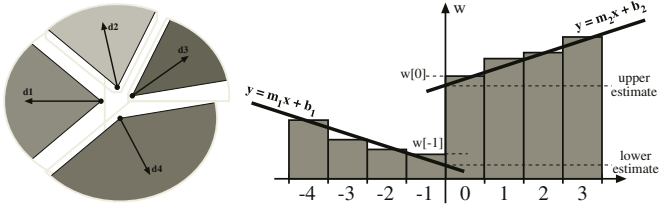
After having obtained the reduced 2D triangle and vertex lists, depth can be attributed to the vertices by using the range data. This is not trivial because (a) the depth may show abrupt changes at discontinuous (jump) edges, (b) the vertex connectivity must be assured at fold edges, and (c) the range data may be very noisy. See Fig. 6 for a standard direct triangulation of a range image, obtained by displaying two triangles at every  $2 \times 2$  pixel block: clearly the data are very noisy.

In a first step, we group the vertices having the same coordinates, which belong to different but neighbouring polygons. Although all polygons have been triangulated, we keep working with polygons because the detection of depth discontinuities is much easier and faster. For each vertex of a group we determine the interior of the polygon it belongs to, and a “search axis” into the polygon by dividing the inner angle by 2 (see Fig. 7 (left)).

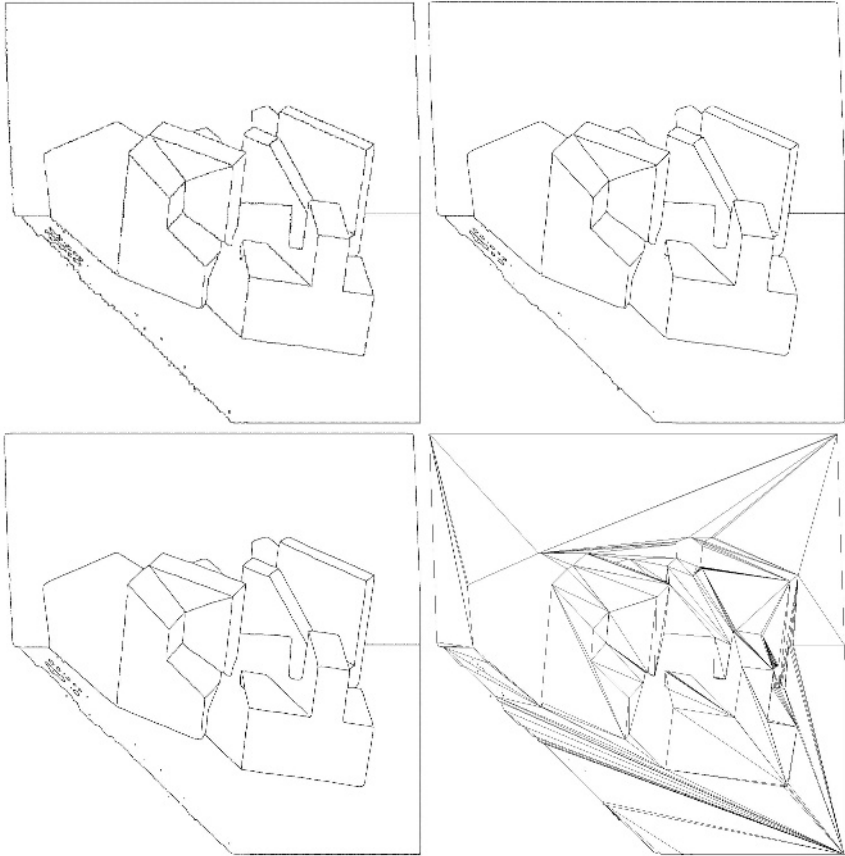
Then, for each group, depth discontinuities are detected along all search axes of the group, using a moving window of size  $2 \cdot S$ ;  $S = 4$  implies a search window of size 8, on which the pixel positions are numbered -4, -3, -2, -1, 0 (the “centre” pixel), 1, 2 and 3, see Fig. 7 (right). On both parts of the search window, the depth information is approximated by linear regression, i.e. on  $[-4, -1]$  and on  $[0, 3]$ . The two depths computed at position  $-0.5$  are compared and ordered, giving  $D_{\min}$  and  $D_{\max}$ , as are the actual depth values at positions -1 and 0,  $D_{-1}$  and  $D_0$ . The depth values at positions -1 and 0 are considered to represent a depth discontinuity if all four of the following conditions hold:

$$|D_0 - D_{-1}| > T, \quad D_{\max} - D_{\min} > T, \quad |D_0 - D_{\min}| > T \text{ and } |D_{-1} - D_{\max}| > T,$$

in which  $T$  is a threshold value. If more than one depth discontinuity is found for a group, only the one closest to the vertex position will be used.



**Fig. 7.** Search axes for a group of vertices (left), and discontinuity detection (right).



**Fig. 8.** Segmented regions after polygonisation, vertex optimisation, path extrapolation at forks and triangulation.

If a depth discontinuity has been found: (a) the depth of each vertex of the group is determined in the direction of its own search axis, but starting at the position of the discontinuity; (b) a search window of 8 pixels is used; (c) the depth along the window is approximated by linear regression; (d) if the difference

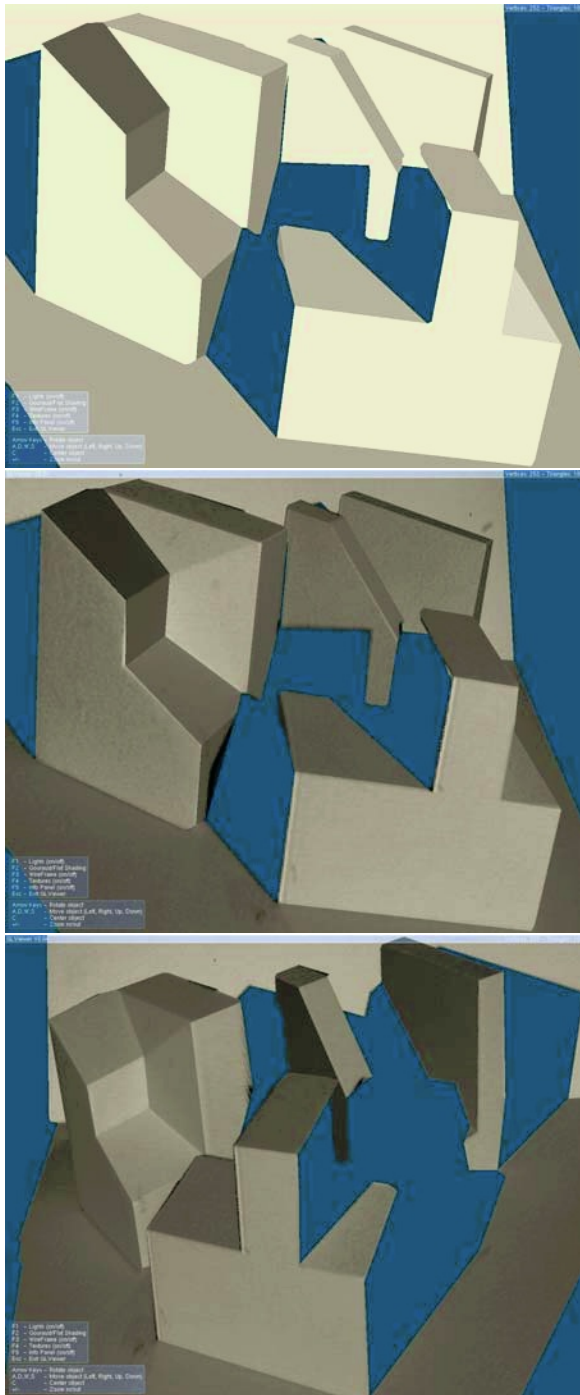


Fig. 9. Different views of a scene with Gouraud shading and texture mapping.

between the linear regression and the actual depth value at each window position is not greater than a threshold, the actual depth values are considered to be valid; (e) the depth of the vertex is calculated by extrapolating the linear regression of the first window position with valid depth values; and (f) after determining the depth of each vertex of the group, vertex connectivity is ensured by comparing depth values (all differences) against a threshold, and by replacing similar values with their corresponding average.

If no depth discontinuity has been found: (a) the depth will be the same for all vertices of the group, since they are considered to be connected; (b) the depth value is obtained directly at the vertex position of the range image; and (c) if, at this position, the depth is not available, we use the average of the first available values along all search axes.

## 6 Results and Discussion

We applied the developed algorithms to the ABW range data set [6]. Average scene CPU times on a 900MHZ iBook with 640MB RAM were 2 to 3s. Triangulation with our algorithms yields up to 1,000 triangles per scene (direct triangulation yields triangle counts up to  $2 \times 511 \times 511 = 522,242$ ). Thus, our algorithms clearly enable interactive object visualisation with shading and texture mapping. Figure 8 shows processing results of the polygonisation, vertex filtering, fork estimation and triangulation for one scene. Note the improved positions of the complex vertex junctions after the fork processing. Figure 9 shows screenshots from our interactive scene visualiser, obtained from two different viewpoints. Note the discontinuity at jump edges and the connectivity at fold edges. The texture mapping (centre and bottom) provides in a more realistic rendering of the scene.

In this paper we presented very efficient, fast and low-level algorithms for the reconstruction of objects in 3D scenes. Further optimisation, currently being explored, concerns the detection of complex vertices (forks) by directly analysing the range data. This is important because, in the case of objects with planar faces, a good *a priori* localisation of forks can save most of the adaptive edge filtering.

## References

1. A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher, "An experimental comparison of range image segmentation algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 7, pp. 673–689, 1996.
2. X. Jiang, K. Bowyer, Y. Morioka, S. Hiura, K. Sato, S. Inokuchi, M. Bock, C. Guerra, R. Loke, and J. du Buf, "Some further results of experimental comparison of range image segmentation algorithms." in *Proc. 15th ICPR*, vol. 4, Barcelona, Spain, September 3-8 2000, pp. 877–881.

3. R. Loke and J. du Buf, "Segmentation of range images in a quadtree." in *1st Iberian Conf. on Pattern Recogn. and Image Analysis (IbPRIA)*, vol. LNCS 2652. Mallorca, Spain: Springer, 2003, pp. 428–436.
4. I. Khalifa, M. Moussa, and M. Kamel, "Range image segmentation using local approximation of scan lines with application to CAD model acquisition," *Machine Vision and Applications*, vol. 13, pp. 263–274, 2003.
5. J. O'Rourke, *Computational Geometry in C*. Cambridge University Press, 1996.
6. J. Min, *Package of evaluation framework for range image segmentation algorithms*. Available at: <http://marathon.csee.usf.edu/range/seg-comp/SegComp.html>: Univ. of South Florida.

# Stereo Vision System with the Grouping Process of Multiple Reaction-Diffusion Models

Atsushi Nomura<sup>1</sup>, Makoto Ichikawa<sup>2</sup>, and Hidetoshi Miike<sup>2</sup>

<sup>1</sup> Faculty of Education, Yamaguchi University  
Yoshida 1677-1, Yamaguchi 753-8513, Japan  
anomura@yamaguchi-u.ac.jp

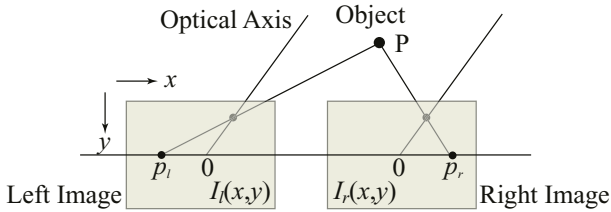
<sup>2</sup> Faculty of Engineering, Yamaguchi University  
Tokiwadai 2-16-1, Ube 755-8611, Japan

**Abstract.** The present paper proposes a system that detects a stereo disparity map from random-dot stereograms with the grouping process. A simple operation for random-dot stereograms converts the stereo correspondence problem to the segmentation one. For solving the segmentation problem derived from random-dot stereograms, the stereo vision system proposed here utilizes the grouping process of our previously proposed model. The model for the grouping process consists of multiple reaction-diffusion models, each of which governs segments having a disparity in the stereo vision system. A self-inhibition mechanism due to strong inhibitory diffusion within a particular reaction-diffusion model and a mutual-inhibition mechanism among the models are built in the proposed system. Experimental results for artificially generated random-dot stereograms show the validity of the proposed system.

## 1 Introduction

In detecting disparity from stereo images, there are two major problems causing disparity error. These are the miss-match problem and the occlusion (un-matched) problem. Most of ordinary methods detect disparity from stereo images by the pattern matching procedure. When a stereo camera system captures a 3-dimensional scene having similar objects or not having texture of brightness patterns, we can not distinguish correspondences of patterns between the stereo images. Similar patterns in the stereo images cause the miss-match problem. When there are two objects located at different distances of depth in 3-dimensional space, one of the objects occludes the other one in the stereo images. The pattern matching procedure can not find the exact correspondence of the occluded object between the stereo images. This is the occlusion problem.

Ordinary methods detect disparity at a particular pixel site by taking account of disparities detected at neighbouring pixel sites. For example, some of the methods propagate disparity over a local neighbouring region with a diffusion process. Since the diffusion process [1] averages the disparity distribution, it also averages abnormal disparities caused by the miss-match problem. The diffusion process also fills in un-matched or un-detected regions by diffusing disparities obtained within the well detected regions. However, since the diffusion process



**Fig. 1.** Geometry of a stereo vision system. A point  $P$  in 3-dimensional space is projected onto the position  $p_l$  on the left image plane  $I_l(x, y)$  and the position  $p_r$  on the right image plane  $I_r(x, y)$ . Optical axes of the two image planes are parallel and horizontal axes of the planes share a common horizontal line.

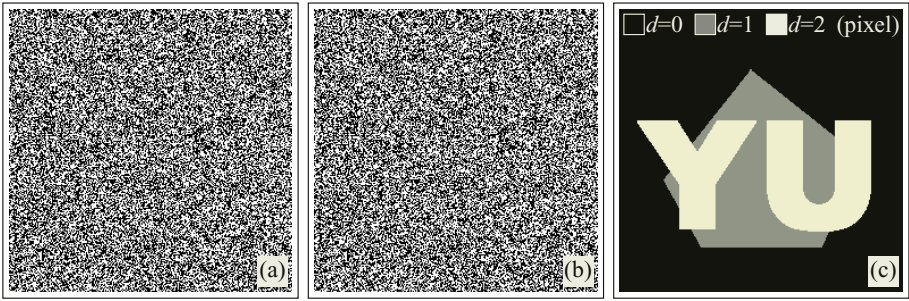
also propagates the disparities across object boundaries and around corners of patterns, it simultaneously causes the over-smoothed problem.

Our previous paper presented a model for the grouping process, which groups the pixel sites having similar features [2]. The model consists of multiple reaction-diffusion models, each of which consists of reaction terms and diffusion ones. The model can suppress the over-smoothed problem, which is often caused by the simple diffusion model, by the self-inhibition mechanism. A special condition on the diffusion coefficients and the non-linear reaction terms of the model prevent the over-smoothed problem. Our another paper also showed that the problem of finding correspondence relation between stereo images becomes the segmentation problem with a simple logic operation [3]. Thus, we are expecting that the model of the grouping process proposed previously solves the segmentation problem derived from the stereo images without the over-smoothed problem.

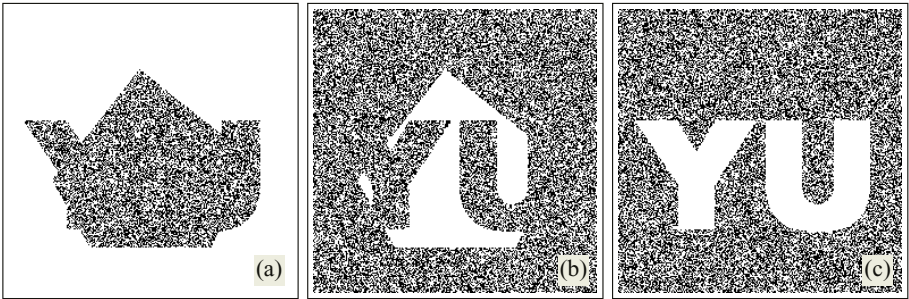
The present paper proposes the stereo vision system detecting a disparity map with the grouping process. The proposed system does not solve explicitly the stereo correspondence problem, but solves the segmentation problem with the grouping process. The main goal of the present study is to avoid the miss-match problem and the occlusion one with the grouping process. The experimental results for artificially generated random-dot stereograms show the performance of the proposed system.

## 2 Stereo Vision System and Random-Dot Stereograms

A stereo vision system captures a 3-dimensional scene through the two cameras located at two different positions (Fig. 1). The system projects a point  $P$  in 3-dimensional space onto the position  $p_l$  of the left image plane  $I_l(x, y)$  and the position  $p_r$  of the right image plane  $I_r(x, y)$ . The stereo disparity refers to difference between the two positions  $p_l$  and  $p_r$  on the horizontal axis. Since the stereo disparity corresponds to the depth one-to-one, it provides the depth of the point  $P$ . For the detection of the disparity, it is necessary to find the correspondence relation between  $p_l$  and  $p_r$  on the stereo images. Most of the previous studies find the correspondence relation by searching similar brightness patterns between the stereo images. They often utilize the pattern matching procedure to obtain the similarity between brightness patterns.



**Fig. 2.** Example of random-dot stereograms. (a) The left image  $I_l(x, y)$  and (b) the right image  $I_r(x, y)$  of the stereo images. The black-dot density of the stereo images is 50(%). (c) The true disparity map, which has three different disparities  $d = 0, 1, 2$  (pixel). The size of the images is  $250 \times 250$  (pixel<sup>2</sup>).

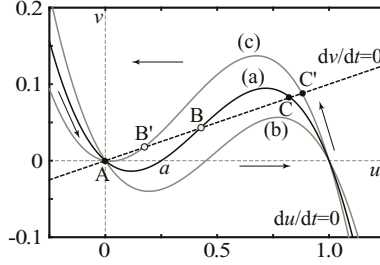


**Fig. 3.** Outputs of the XNOR logic operation applied to the random-dot stereograms (Fig. 2). In the stereograms, a white dot refers to the logical value “true” and a black dot does to “false”. Overlapping the stereo images [Figs. 2(a) and 2(b)] located at a difference  $d = 0, 1, 2$  (pixel) and computing the XNOR logic operation for the stereo images at a particular pixel site provided the three outputs  $L(x, y; d)$  [see Eq.(1)]. (a)  $L(x, y; d = 0)$ , (b)  $L(x, y; d = 1)$  and (c)  $L(x, y; d = 2)$ .

The random-dot stereograms show that the human visual system can perceive the depth from only the disparity information (Fig. 2) [4]. The random-dot stereograms have only dot patterns; corresponding dots are located at slightly different positions in the stereo images. The difference of the corresponding positions of a dot in the stereo images is the disparity. Finding the correspondence of the dot between the stereo images provides the disparity, which provides the depth.

When we focus on the random-dot stereograms, the problem of finding the correspondence relation between the stereo images becomes the segmentation problem [3]. Let us suppose that one of the stereo images overlaps the other one, where the centre of the former image plane differs from that of the other image plane. The distance between the two centre positions is  $d$  (pixel) on the horizontal axis. We suppose that a black-or-white value at a pixel site in a random-dot image refers to a logical value of “false” or “true”. The output  $L(x, y; d)$  of the XNOR logic operation,





**Fig. 4.** Phase plot for ordinary differential equations  $du/dt = f(u, v)/\varepsilon$  and  $dv/dt = g(u, v)$  with the FitzHugh-Nagumo type reaction terms of Eq.(3). The parameter  $a$  is (a)  $a = 0.25$ , (b)  $a = 0.45$  and (c)  $a = 0.05$ ; the parameter  $b$  is fixed as  $b = 10$ . The points A, C and C' are stable steady states; the points B and B' are unstable ones. The systems (a) and (c) having two stable steady states is called the bi-stable system; the system (b) having one stable steady state is called the mono-stable system.

$$L(x, y; d) = \overline{I_l(x, y) \oplus I_r(x + d, y)}, \quad (1)$$

applied to two dots on the overlapped stereo images  $I_l(x, y)$  and  $I_r(x, y)$  extracts the region having the disparity  $d$  (pixel) as the flat white pattern having the true value. The random-dot pattern remains in other regions not having the disparity  $d$ . Thus, the segmentation of the white flat region from the random-dot pattern region for  $L(x, y; d)$  extracts the region having the disparity  $d$  (Fig. 3).

### 3 Proposed Stereo Vision System

#### 3.1 FitzHugh-Nagumo Type Reaction-Diffusion Model

A general reaction-diffusion model with two variables  $u$  and  $v$  consists of two partial differential equations describing the temporal developments of the two variables. The equations have diffusion terms of  $\nabla^2 u$  and  $\nabla^2 v$  and reaction ones  $f(u, v)$  and  $g(u, v)$ ,

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u + \frac{1}{\varepsilon} f(u, v) + \mu s, \quad \frac{\partial v}{\partial t} = D_v \nabla^2 v + g(u, v), \quad (2)$$

where  $D_u$  and  $D_v$  are diffusion coefficients,  $s(x, y)$  is a source term and its coefficient  $\mu$  is a small constant ( $0 < \mu \ll 1$ ). The FitzHugh-Nagumo type reaction terms [5, 6] refer to the next functions,

$$f(u, v) = u(1 - u)(u - a) - v, \quad g(u, v) = u - bv, \quad (3)$$

where  $a$  and  $b$  are constants.

Figure 4 shows the trajectory of the solution  $(u, v)$  under the non-diffusive system ( $D_u = D_v = 0$ ) and without the source term ( $s = 0$ ). When the model is the bi-stable system, a solution converges either of two stable steady states; when the model is the mono-stable system, a solution converges a stable steady state. The system becomes either of the mono-stable system or the bi-stable one, according to the parameter values of  $a$  and  $b$ . In addition, the parameter  $a$

works as a kind of a threshold value for an initial solution of  $(u = u_0, v = 0)$ . When  $u_0 < a$ , the solution directly converges to the stable steady state A at the origin  $(u = 0, v = 0)$ . When  $u_0 > a$ , the solution first moves toward the point  $(u = 1, v = 0)$  along the horizontal coordinate. After that, if the system is bi-stable, the solution finally converges to the stable steady state C or C' along the function  $du/dt = 0$ ; if the system is mono-stable, the solution does to the stable steady state A along the trajectory indicated by the arrows in Fig. 4.

### 3.2 Multiple Reaction-Diffusion Models and Grouping Process

The next set of equations having two variables  $(u_d, v_d)$  describes the modified version of the FitzHugh-Nagumo type reaction-diffusion model,

$$\frac{\partial u_d}{\partial t} = D_u \nabla^2 u_d + \frac{1}{\varepsilon} f(u_d, v_d, u_m) + \mu s_d, \quad \frac{\partial v_d}{\partial t} = D_v \nabla^2 v_d + g(u_d, v_d), \quad (4)$$

where the output  $L(x, y; d)$  of the XNOR logic operation for the disparity  $d$  is provided to the source term  $s_d(x, y)$ . The set of equations Eq.(4) governs the groups having the disparity  $d$ . The disparity is in the range of  $d = 0, 1, 2, \dots, D$ . Thus, multiple models, the number of which is  $D + 1$ , are necessary to govern the multiple disparity values. The parameter  $u_m$  refers to the maximum value of  $u_d$ , namely,  $u_m = \max_d(u_d)$ .

We introduce the mutual-inhibition mechanism among the multiple reaction-diffusion models, each of which governs the groups of the disparity  $d$ . We call the state having the high value of  $u_d \simeq 1$  "excited". Let us consider that the pixel site being the excited state has the disparity  $d$ . When a model becomes the excited state, since a particular pixel site has only one disparity value, the other models must not become the excited state. The original FitzHugh-Nagumo model has the parameter  $a$ , which works as a threshold value. In order to exclusively detect a disparity value at a pixel site, we introduce the switching function into the parameter  $a$ . Our previous paper [2] proposed the next modified version of the reaction terms,

$$f(u_d, v_d, u_m) = u_d(1 - u_d)(u_d - a(u_m)) - v_d, \quad g(u_d, v_d) = u_d - bv_d, \quad (5)$$

and the next function  $a(u_m)$ ,

$$a(u_m) = \frac{1}{4} \{ \tanh(u_m + a_0) + 1 \}, \quad (6)$$

where  $a_0$  is a constant. When another model becomes the excited state ( $u_m$  becomes large), the threshold value  $a(u_m)$  also becomes large. Therefore, the large threshold value inhibits the model governing the disparity  $d$  from becoming the excited state. This is the mutual-inhibition mechanism built in the modified reaction-diffusion models.

A special condition for the ratio between the diffusion coefficients  $D_u$  and  $D_v$  causes the self-inhibition mechanism in the reaction-diffusion model. In the condition of  $D_v/D_u < 1$ , spatial distributions of  $u_d$  and  $v_d$  change as time proceeds. Edges of spatial patterns in their distributions propagate; their global structures

dynamically change. In the condition of  $D_v/D_u > 1$ , the strong diffusion of  $v_d$ , compared to that of  $u_d$ , inhibits the edges from propagating (the self-inhibition mechanism). Our model utilizes the condition for the self-inhibition mechanism to sustain static patterns expressing the groups of the disparity  $d$ .

### 3.3 Building a Disparity Map

The final stage to build a disparity map is the integration of the outputs  $u_d$  of the multiple reaction-diffusion models. When a model with  $u_d$  is the excited state at a pixel site, we can understand that the pixel site has the disparity  $d$ . Thus, we detect a disparity value at a pixel site by searching the maximum value for all of the outputs  $u_d$ . The proposed system builds the disparity map  $M(x, y, t)$  by,

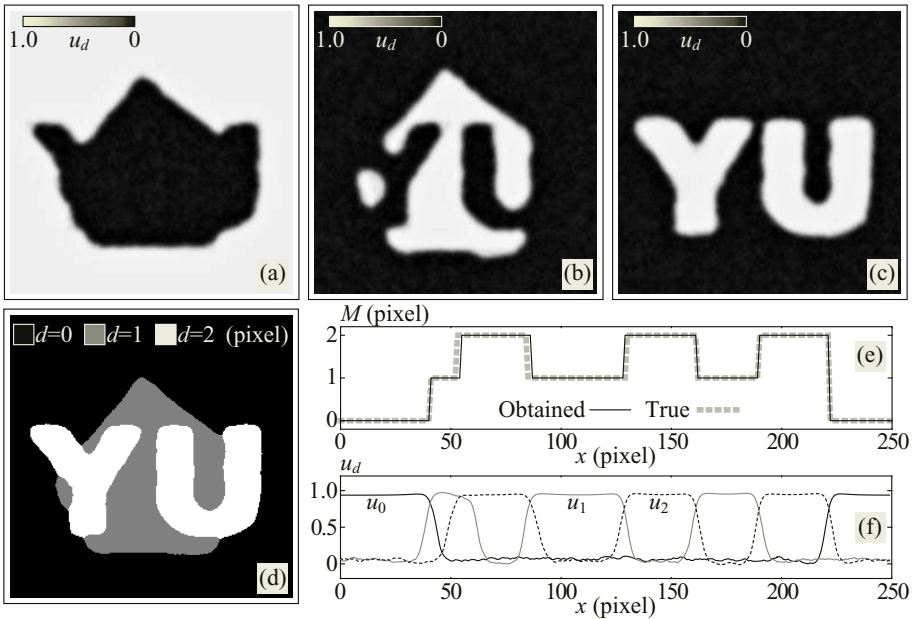
$$u_m = \max_d(u_d) \Rightarrow M(x, y, t) = m. \quad (7)$$

## 4 Experimental Results

We realized the proposed stereo vision system by numerical methods. The finite difference method discretized the partial differential equations of the proposed multiple reaction-diffusion models of Eq. (4). The Crank-Nicolson method with 5 spatial neighbouring points approximates the Laplacian operator  $\nabla^2$ . The Gauss-Seidel method solves the set of linear equations obtained by the discretization. The Neumann boundary condition governs the four sides of the image plane. Initial conditions of  $u_d$  and  $v_d$  are zero over the image plane.

Figure 5 shows the result for the random-dot stereo images of Figs. 2(a) and 2(b). The outputs of Fig. 3 represent the source terms of Eq.(4). Figures 5(a), 5(b) and 5(c) show the distributions of  $u_d$  for  $d = 0, 1, 2$ , respectively. Equation 7 built the disparity map of Fig. 5(d) from the distributions of  $u_d$ . By comparing the result of the disparity map Fig. 5(d) with the true one of Fig. 2(c), we successfully detected the disparity map. For confirming the validity of the obtained map more quantitatively, we showed the 1-dimensional profiles of the obtained disparity map and the true one in Fig. 5(e). We can confirm that these disparity profiles are almost the same except for those around  $x = 50$ . Figure 5(f) shows the 1-dimensional profiles of  $u_d$ . We can confirm that the variables  $u_d$  are almost exclusively distributing in the 1-dimensional space. However, around  $x = 50$ , both the variables  $u_1$  and  $u_2$  become excited. This caused the disparity error around  $x = 50$ .

An additional experiment shows the performance of the proposed method for random-dot stereograms having low dot density. Figure 6 shows the result for the stereo images having the black-dot density of 10(%) [the true disparity map is the same as Fig. 2(c)]. In the outputs of the XNOR logic operation applied to the low density stereo images [Figs. 6(c)~6(e)], there exists many pixel sites having the logical true values (white pixel), compared to those of Fig. 3. Therefore, the problem of finding the flat true regions becomes more difficult. Figure 6(f) shows the disparity map obtained by the proposed system. The shapes included in the obtained disparity map incompletely illustrate the original ones. However, the global structure of the map is very similar to the true one.



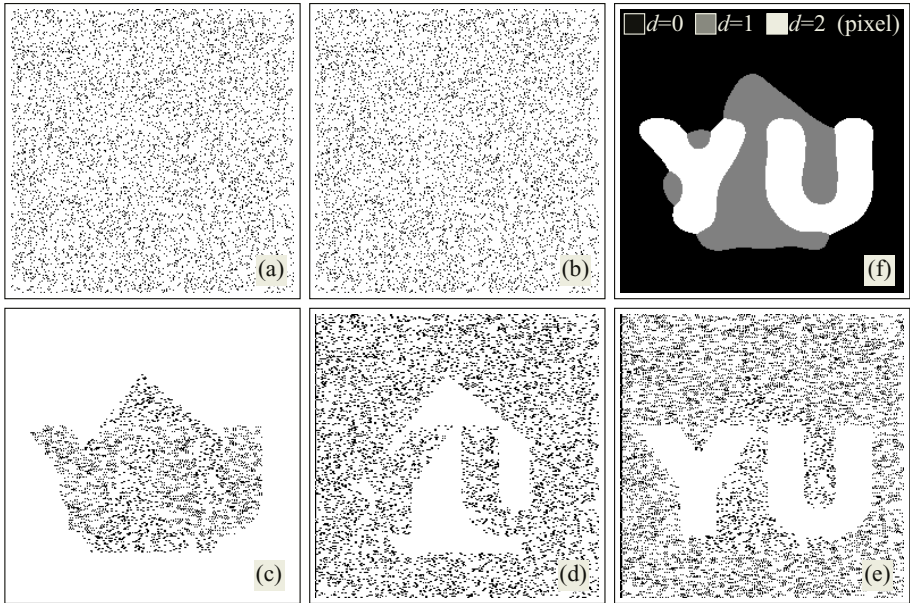
**Fig. 5.** Experimental result for the random-dot stereograms of Fig. 2. The grouping process of the proposed system analysed the outputs of the XNOR logic operation (Fig. 3) and provided the distribution maps of  $u_d$  for (a)  $d = 0$ , (b)  $d = 1$  and (c)  $d = 2$  at  $t = 50$ . The parameter values utilized in the present experiments were  $D_u = 1, D_v = 4, \varepsilon = 1/100, a_0 = 0.25, b = 10, \mu = 0.005$ ; the finite differences were  $\delta x = \delta y = 1/10, \delta t = 1/1000$ . (d) The disparity map  $M(x, y, t)$  obtained from the distributions of  $u_d$ . (e) The 1-dimensional profiles of the obtained disparity map compared with its true map. (f) The 1-dimensional profiles of  $u_d$  for  $d = 0, 1, 2$ .

## 5 Conclusions

The present paper proposed the stereo vision system detecting a disparity map from random-dot stereograms. The problem of detecting stereo disparity becomes the segmentation problem by a simple logic operation for the stereo images. In solving the segmentation problem, the proposed system utilizes the grouping process realized with the multiple reaction-diffusion models having the mutual-inhibition mechanism and the self-inhibition one. The integration of the outputs of the multiple models provides a disparity map. Through the analysis of random-dot stereograms, the validity of the proposed system was confirmed.

## Acknowledgement

The present study was partly supported by the Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science and Technology, Japan (No.15340125).



**Fig. 6.** Experimental results for the random-dot stereograms having low dot density. The black-dot density of the images is 10%. (a) Left image and (b) right image. The size of the stereo images is  $250 \times 250$  (pixel<sup>2</sup>). The XNOR logic operation applied to the stereo images provided the outputs for (c)  $d = 0$ , (d)  $d = 1$  and (e)  $d = 2$  (pixel). (f) The disparity map  $M(x, y, t)$  obtained at  $t = 50$ . The parameter values utilized here were the same as those of Fig. 5 except for  $\mu = 0.003$ .

## References

1. Koenderink, J. J.: The Structure of Images. *Biol. Cybern.* **50** (1984) 363–370
2. Nomura, A., Ichikawa, M., Miike, H.: Realizing the Grouping Process with the Reaction-Diffusion Model. *IPSJ Trans. Computer Vision and Image Media* **45** (2004) 26–39 (in Japanese)
3. Nomura, A., Ichikawa, M., Miike, H., Ebihara, M., Mahara, H., Sakurai, T.: Realizing Visual Functions with the Reaction-Diffusion Mechanism. *J. Phys. Soc. Jpn.* **72** (2003) 2385–2395
4. Julesz, B.: Binocular Depth Perception of Computer-Generated Patterns. *The Bell System Tech. J.* **39** (1960) 1125–1162
5. FitzHugh, R.: Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophysical J.* **1** (1961) 445–466
6. Nagumo, J., Arimoto, S., Yoshizawa, S.: An Active Pulse Transmission Line Simulating Nerve Axon. *Proc. IRE* **50** (1962) 2061–2070

# Registration of Moving Surfaces by Means of One-Shot Laser Projection\*

Carles Matabosch<sup>1</sup>, David Fofi<sup>2</sup>, Joaquim Salvi<sup>1</sup>, and Josep Forest<sup>1</sup>

<sup>1</sup> University of Girona, Institut d'Informàtica i Aplicacions, Girona, Spain

<sup>2</sup> University of Burgundy, Laboratoire d'Electronique, Informatique et Image  
Le Creusot, France

**Abstract.** The acquisition of three-dimensional models of a given surface is a very interesting subject in computer vision. Most of techniques are based on the use of laser range finders coupled to a mechanical system that scans the surface. These techniques lacks of accuracy in the presence of vibrations or non-controlled surface motion because of the misalignments between the acquired images. In this paper, we propose a new one-shot pattern which benefits from the use of registration techniques to recover a whole surface in the presence of non-controlled motion.

## 1 Introduction

Three-dimensional reconstruction of real objects is a promising subject with many applications, such as reverse engineering, robot navigation, mould fabrication and visual inspection among others. Most range finders are based on the projection of laser beams because of its robustness against ambient light, easy image processing algorithms and high given accuracy including optical segmentation and subpixel accuracy. Please, check a quite recent survey related to laser projection [3] and other reconstruction techniques such as coded structured light [9]. In general, laser projection techniques are based on the use of a laser emitter coupled to a cylindrical lens that spread the light forming a plane that is projected to the measured surface. The projection of a laser plane only lets us to reconstruct a profile of the measuring surface. So, in most cases a mechanical system is added that permits a scanning. In some applications: a) the laser plane is projected onto a rotating mirror and reflected towards the surface; b) the laser beam is attached to a moving worm gear; c) the laser beam keeps motionless while is the object which is placed on a rotating table. All these techniques permit the reconstruction of a whole surface with high resolution. However, the accuracy strongly depends on the mechanical system used so that potential vibrations given by the environment produces misalignments and consequently the accuracy is considerably influenced. Furthermore, the sequence of images that are captured in the scanning process forces the object to be motion controlled reducing the number of applications, i.e. industrial conveyors can not be considered.

In this paper, a new one-shot 3D sensor is proposed, which is based on registering a set of 3D images from a non-controlled moving surface. Furthermore, dense cloud of

---

\* This work is partially supported by the Spanish Project TIC2003-08106-C02-02.

3D points are acquired without using any mechanical system to scan the object so that misalignments in the reconstruction are neglected. Although one-shot 3D sensors have been previously used, usually a manual or mechanical process is required to align the scanned surfaces [7]. In this paper, a pair-wise-based registration method is proposed to align the cloud of 3D points with the aim of obtaining a complete surface of the scanned object.

## 2 One-Shot 3D Sensor

Nowadays, there are a considerably amount of lenses which can be coupled to a laser emitter which spreads the light forming a given pattern: planes, circles, dots and stripes. However, it has been demonstrated that stripe patterns are the most suitable in measuring processes because of the easy segmentation and the use of subpixel techniques in the detection of the stripe peaks. Stripe patterns also ease the search of correspondences among the slits projected and the ones acquired by the camera. The number of stripes projected is directly related to the surface resolution and to the image processing complexity. A compromising stripe pattern forming 19 slits has been chosen and the images are acquired by a on-the-shelf camera coupled with a 635 nm optical filter.

## 3 Calibration

Calibration is an offline process which aim is the computing of the geometry that relates the 3D points on the measuring surfaces with the projection of these points in the acquired image. This relation can be linearly approximated to the following equation:

$$P_W = {}^W T_L \cdot p_i \quad (1)$$

Once  ${}^W T_L$  is known, 2D points in the image frame can be directly transformed to 3D points in the world reference frame. This matrix is computed by orthogonal least squares from a set of correspondences, also known as calibrating points. In order to search for correspondences, the complete quadrangle is used [2]. The original method has been adapted to calibrate the set of 19 planes obtaining the 19 transformation matrices which describes the geometry of the sensor. For every plane calibration the following steps are processed:

- Detection of the points of the laser profile in the image plane,
- Find the correspondences between points in the image plane and 3D points in the calibrating plane,
- and Compute the T matrix using the correspondences given by the previous step.

In the following sections, the three steps are described.

### 3.1 Points in the Laser Profile

When a unique plane is projected to the scene, the peak detection with subpixel accuracy can be determined with high accuracy using a FIR filter approach [4]. However,

when more planes are projected, the derived curve of the profile (shown in fig.1b) is high influenced by the neighborhood. In some situations, the derived curve does not cross to zero at the maximum value of the intensity profile. To solve this problem, an adapted methodology that is based on a previous work related to coded structured light is used [8]. First of all, the first derivative is computed using the convolution of each row with the vector  $[-1 -1 -1 0 1 1 1]$ . Then, the second derivative is computed obtaining the enhancement of the peaks compared to the intensity image. A threshold is finally used to segment the stripes as follows:

$$\begin{cases} 0 & \text{if } f_i'' < \text{mean}(f) + \text{var}(f) \\ 255 & \text{otherwise} \end{cases} \quad (2)$$

$$f_L' = \text{conv}([1 -1 -1 0 1 1 1], [f(p_i - 3) : f(p_i + 3)]) \quad (3)$$

where  $f$  is the intensity profile curve and  $f_i''$  is the second derivative in each pixel of the row. As can be seen in the fig. 1c, the interval of each peak can be found easily analyzing all the pixels in a consecutive order. For each interval, the central value is computed as an approximation of the position of each maximum. Then, a local derivative is computed in each estimated peak as follows:

where  $\text{conv}$  is the convolution, and  $f(p_i)$  is the value of the intensity profile in the  $i$ th estimated peak. The pass to zero of the  $f_L'$  function give us the sub-pixel position of the peak of each laser stripe. Furthermore, if the intensity value of this points is less than a threshold, this peak is not considered.

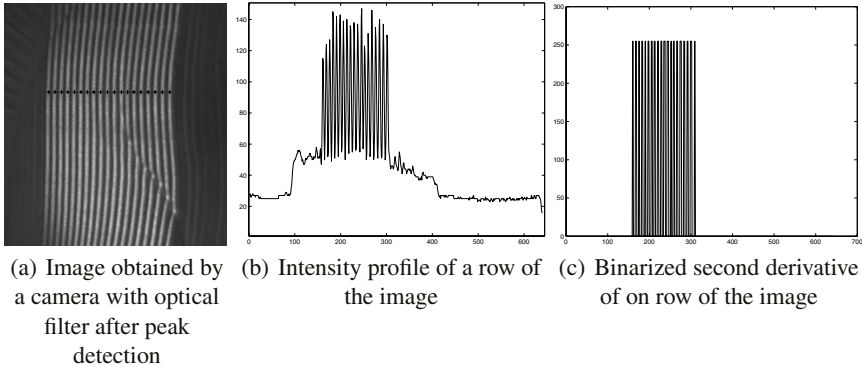


Fig. 1. Process of obtaining the laser peaks

### 3.2 Correspondences Between Points in the Image and 3D Points

The methodology is based on the *complete quadrangle* [1]. The principle of this method is the cross-ratio between the complete quadrangle and the acquired image of this quadrangle (see fig. 2).

$$\frac{\overline{A'P'_A}}{\overline{A'G'}} = \frac{\overline{AP_A}}{\overline{AG}} \quad (4)$$



As  $A, B$  are known 3D points, and  $A', B'$  and  $P'_A$  can be found analyzing the acquired image,  $P_A$  can be determined. The same principle is applied with point  $P_B$ . If quadrangle is moved along the  $Z$ -axis, a set of 2D-3D correspondences can be found for each  $Z$  position. Using this set of correspondences, eq. 1 can be solved determining the transformation matrix. In general, only two points are used for every plane position. Note that calibration accuracy is related directly to the number of correspondences used. In order to improve the accuracy, a set of points along the laser stripe are selected. More details are presented in [2].

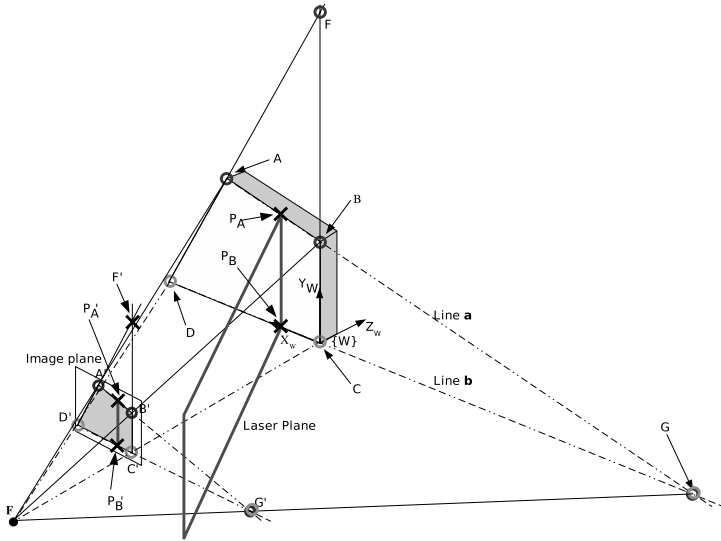


Fig. 2. Cross-ratio and the complete quadrangle used to determine 2D-3D correspondences

### 3.3 Compute T Matrix Using Known Correspondences

Now the transformation matrix can be obtained by minimizing eq. 5 which has been easily obtained arranging eq. 1.

$$\begin{bmatrix}
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 u_i & v_i & 1 & 0 & 0 & 0 & 0 & 0 & -u_i \cdot X_i & -v_i \cdot X_i & -X_i \\
 0 & 0 & 0 & u_i & v_i & 1 & 0 & 0 & -u_i \cdot Y_i & -v_i \cdot Y_i & -Y_i \\
 0 & 0 & 0 & 0 & 0 & 0 & u_i & v_i & 1 & -u_i \cdot Z_i & -v_i \cdot Z_i & -Z_i \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
 \end{bmatrix} \cdot \begin{bmatrix} t_{11} \\ t_{12} \\ t_{13} \\ t_{21} \\ \vdots \\ t_{43} \end{bmatrix} = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (5)$$

where  $t_{ij}$ 's are the parameters of the  ${}^W T_L$  matrix. The solution is obtained from the computation of the vector  $\theta$  that minimizes equation  $A \cdot \theta = 0$ . A good estimation using Orthogonal Least Square technique is computed from the eigenvector corresponding to the smaller eigenvalue of matrix  $A^T \cdot A$ .

## 4 Reconstruction

Once the system is calibrated and the transformation matrices for every stripe computed, the 3D points can be reconstructed by using their corresponding transformation matrix. So, next step in reconstruction is stripe segmentation and the correspondence problem. A robust stripe identification has been implemented which label every stripe when all them are present for a given image row [3]. This information is used as a seed to complete the stripe identification by region growing that allows us to identify the stripes in the presence of occlusions and cuts. Then, once every image pixel is labelled to the corresponding stripe, the surface reconstruction is accomplished.

A further step deals with the interpolation of the 3D profiles obtained with the aim of obtaining a continuous surface. The function used to approximate the surface is the following:

$$z = ax^2 + by^2 + cxy + dx + ey + f \quad (6)$$

The parameters are obtained by Least Squares as follows:

$$\begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = (H^T H)^{-1} H^T \begin{pmatrix} z_1 \\ \vdots \\ z_i \\ \vdots \\ z_n \end{pmatrix} \text{ where } H = \begin{pmatrix} x_1 x_1 & y_1 y_1 & x_1 y_1 & x_1 & x_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n x_n & y_n y_n & x_n y_n & x_n & x_n & 1 \end{pmatrix} \quad (7)$$

The results of the reconstruction are shown in fig 3. In spite of only 19 planes are used to acquire the surface, the resolution of the final reconstruction is enough in free-form shape objects. Furthermore, details not acquired by the sensor can be obtained in the registration process, where some partial views are fused.

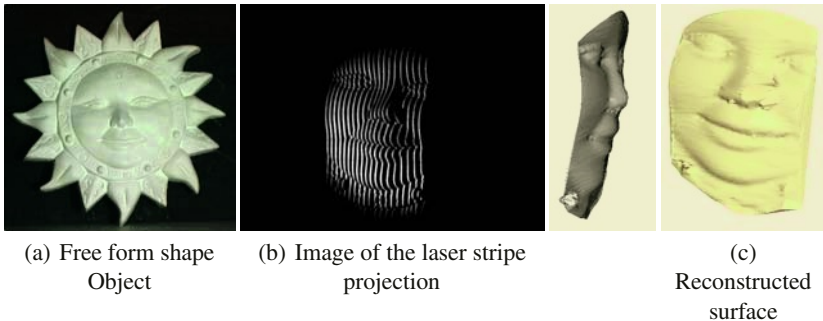
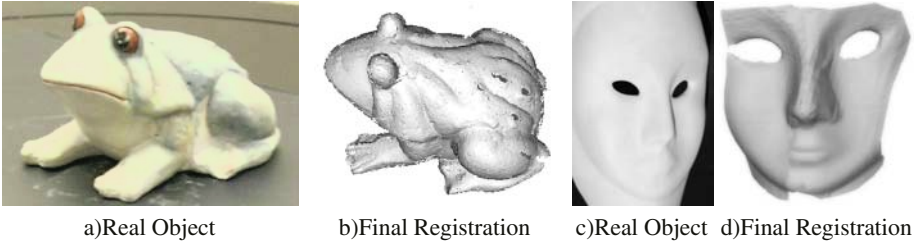


Fig. 3. Experimental results with a real object

## 5 Registration

When a set of free views from a given object are already available, registration can be applied to align all these views among them with respect to a reference system and



**Fig. 4.** Final Results

**Table 1.** Registration results of the frog using real data

Real angle	Rotation angle error			
	Pair-wise (ICP)		Our proposal	
	Computed angle	Error	Computed angle	Error
45°	44.11°	0.89°	44.11°	0.89°
90°	88.41°	1.59°	88.41°	1.59°
135°	132.92°	11.08°	124.20°	10.80°
180°	168.60°	11.40°	183.86°	3.86°
225°	213.00°	12.00°	228.27°	3.27°
270°	256.39°	13.61°	271.67°	1.67°
315°	300.75°	14.25°	316.03°	1.03°

obtain a complete reconstruction of the object. A state-of-art of Registration methods has been recently published [6]. The results of this work pointed out that the best technique to register range images is a robust variant of ICP [10] which was classified as a pair-wise registration technique. Once all the images have been registered in pairs using Zimmer method, a global minimization is applied with the aim of reducing the global error. A graph of connectivity is constructed analyzing if two views are connected by a common surface region. The goal is to compute the transformation of each view to the reference frame throughout the path in the graph with minimal residual error, where the error is computed as the mean of the distances between point correspondences for every pair of views [5]. Dijkstra algorithm is applied to determine the optimal path in graphs to solve this problem, obtaining a reduced graph. At last, the paths with minimal error are the ones used to register the set of views and the object reconstruction is completed. Figure 4b shows an example of the registration of 8 different views of an object where the images has been captured by using a Minolta Vivid 700 Scanner and the object where placed on a non-controlled rotating table. This figure evaluates the accuracy of the proposing registration method. In table 1, the rotation error obtained is compared with the results of traditional pair-wise without refinement. Furthermore, figure 4d shows the results of the registration of ten views captured by the one-shot scanner proposed. Obviously, reconstructions are not as accurate as the Minolta equipment, but note that the proposed scanner captures 3D information in a single image and moreover the registration can be refined by the capturing of more and more views of the same object.

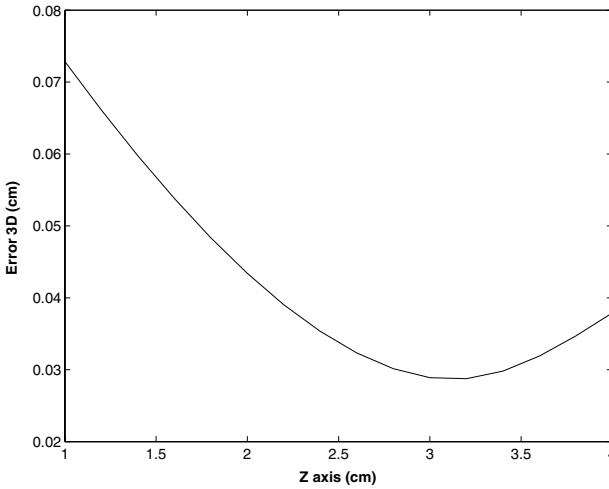


Fig. 5. The accuracy obtained related to depth, i.e. the Z-axis

## 6 Experimental Results

A set-up consisting of one on-the-shelf CCD camera, a 635 nm LASIRIS laser emitter and an optical lens which spreads the laser beam into 19 planes has been arranged conforming the imaging system. Both camera and laser are located on a portable platform where their optical axis form an angle of  $60^\circ$  and the distance between them is approximately 20cm. A calibrating quadrangle has been located at several distances from the system in increments of 2 mm. The closest plane is located at 20 cm. from the imaging system. For every quadrangle position, two images are acquired: a) the first is an image of the quadrangle; b) the second is the projection of the laser on the quadrangle. The first image is used to determine the parameters of the quadrangle while the second the geometry of the laser. Then, every laser stripe is determined by a sequence of 16 correspondences which are used to compute the transformation matrix for each stripe. The accuracy of the system is computed from the discrepancy between the reconstructed 3D points and the 3D points used in the calibration process. The results are shown in fig. 5. The error is represented with respect to Z-axis which is the axis more sensitive and directly related to depth. The results give a good accuracy in a narrow area covered the center of the calibration area while the accuracy decreases in the vicinity.

## 7 Conclusions

This paper presents a new one-shot imaging system, which is based on a single on-the-shelf camera and a stripe laser pattern. The system benefits from one-shot techniques to recover the 3D shape of surfaces in non-controlled motion environments or even in the presence of vibrations. Registration is used to align every 3D acquisition with respect to a world coordinate system obtaining a complete reconstruction of the measuring object. The calibration benefits from the use of the complete quadrangle and image processing

from the use of a nice stripe peak detector with subpixel accuracy. Experimental results show that the accuracy obtained in the reconstruction step is quite acceptable (less than 0.5 mm. in the centered area) and the visual quality of registered surface satisfactory.

## References

1. C. Chen and A. Kak. Modelling and calibration of a structured light scanner for 3d robot vision. In *IEEE conference on robotics and automation*, pages 807–815, 1987.
2. J. Forest. *New methods for triangulation-based shaped acquisition using laser scanners*. PhD thesis, University of Girona, 2004.
3. J. Forest and J. Salvi. An overview of laser slit 3d digitasers. In *International Conference on Robots and Systems*, pages 73–78, Lausanne, October 2002.
4. J. Forest, J. Salvi, and C.; Cabruja, E. and Pous. Laser stripe peak detector for 3d scanners. a fir filter approach. In *International Conference on Pattern Recognition*, volume 3, pages 646 – 649, Cambridge, United Kingdom, August 2004.
5. C Matabosch, J. Salvi, and D. Fofi. A new proposal to register range images. In *7th International Conference on Quality Control by Artificial Vision*, Nagoya, Japan, May 2005.
6. C. Matabosch, J. Salvi, X. Pinsach, and R. García. Surface registration from range image fusion. In *IEEE International Conference on Robotics and Automation*, New Orleans, April-May 2004.
7. P. Mueller, T. Vereenooghe, M. Vergauwen, L. Van Gool, and M. Waelkens. Photo-realistic and detailed 3d modeling: the antonine nymphaeum at sagalassos (turkey. In *Computer Applications and Quantitative Methods in Archaeology (CAA): Beyond the artifact - Digital interpretation of the past*, Prato, Italy, April 2004.
8. J. Pagès and J. Salvi. A new optimised de bruijn coding strategy for structured light patterns. In *17th International Conference on Pattern Recognition*, volume 4, pages 284 – 287, Cambridge, United Kingdom, August 2004.
9. J. Salvi, J. Pagès, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, April 2004.
10. T. Zimber and H. Schmidt, J. Niermann. A refined icp algorithm for robust 3-d correspondences estimation. In *International Conference on Image Processing*, pages 695–698, September 2003.

# A Computer Vision Sensor for Panoramic Depth Perception\*

Radu Orghidan<sup>1</sup>, El Mustapha Mouaddib<sup>2</sup>, and Joaquim Salvi<sup>1</sup>

<sup>1</sup> Institute of Informatics and Applications, Computer Vision and Robotics Group  
University of Girona, Girona, Spain  
{radu, qsalvi}@eia.udg.es

<sup>2</sup> Centre de Robotique, Électrotechnique et Automatique  
Université de Picardie Jules Verne, Amiens, France  
mouaddib@u-picardie.fr

**Abstract.** A practical way for obtaining depth in computer vision is the use of structured light systems. For panoramic depth reconstruction several images are needed which most likely implies the construction of a sensor with mobile elements. Moreover, misalignments can appear for non-static scenes. Omnidirectional cameras offer a much wider field of view than the perspective ones, capture a panoramic image at every moment and alleviate the problems due to occlusions. This paper is focused on the idea of combining omnidirectional vision and structured light with the aim to obtain panoramic depth information. The resulting sensor is formed by a single catadioptric camera and an omnidirectional light projector.

## 1 Introduction

The omnidirectional vision sensors enhance the field of view of traditional cameras by means of special optics, structures of still or gyrotory cameras or combinations of lenses and mirrors. Yagi [14] surveyed the existing techniques for building cameras with a wide field of view and Svoboda [13] proposed several classifications of the existing omnidirectional cameras according to their most important features.

The catadioptric sensors use reflecting surfaces (convex or planar mirrors) coupled to a conventional camera and are usually classified depending on the way they gather the light rays. When all the observed light rays converge into a point, called focus, the sensors are known as Single View Point (SVP) [1]. The SVP enables distortion-free reconstruction of panoramic images in a familiar form for the human users.

Stereo catadioptric sensors are special structures of mirrors and lenses designed for obtaining depth from images with a wide field of view. In order to obtain distinct points of view of the scene the camera is pointed towards a structure of convex [3] or planar [5] mirrors. The results obtained by stereoscopic vision depend on the accuracy of matching the points between the observed images. Structured light based techniques

---

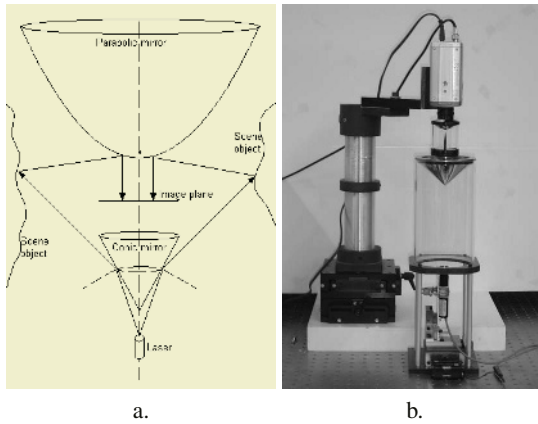
\* This work is partially supported by the Spanish project CICYT TIC 2003-08106-C02-02 and by the AIRE mobility grant provided by the Generalitat of Catalunya that allowed a four month stay in the CREA lab from Amiens, France.

are a particular case of stereo vision where one of the cameras is replaced by a pattern projector [12]. Using this technique is similar to placing visible landmarks in the scene so that image points can be identified and matched faster.

This paper presents an omnidirectional sensor that provides 3D information using structured light. The sensor is formed by a single-camera catadioptric configuration with an embedded omnidirectional structured light projector. By mounting the omnidirectional sensor on a mobile robot applications such as 3D map building, robot navigation and localization, active surveillance with real-time object detection or 3D reconstruction can be performed within a horizontal field of view of 360 degrees. The sensor design and the calibration of the whole system is detailed in section 2. The experimental results are shown in section 3. The article ends with conclusions, presented in section 4.

## 2 Sensor Geometry

In the proposed solution, see Figure 1, the omnidirectional camera is coupled with a structured light projector that has a field of view of 360 degrees. A more compact sensor can be built by placing the light projector within the blind zone of the omnidirectional camera as shown in [8] where a similar sensor was described and analyzed by simulation. However, for the realization of the first prototype of the physical sensor the two parts have been separated for more maneuverability.



**Fig. 1.** a. Catadioptric omnidirectional camera with embedded structured light projector. b. Laboratory prototype.

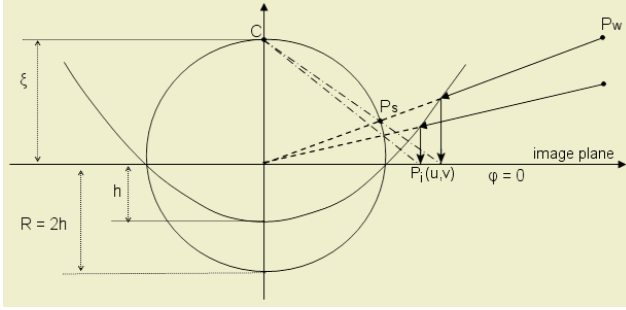
The circular pattern projected by the laser is reflected by the conical mirror and becomes a light-stripe on the scene. The parabolic mirror reflects the scene into the camera and the laser-stripe can be immediately identified. With the models for both components of the sensor a precise triangulation can be carried out.

The traditional approach for calibrating a structured light system takes two steps. The camera is calibrated at first and the light projector is subsequently calibrated based

on information provided by the camera. A method based on the cross ratio invariance under perspective projection providing a direct image to world transformation was proposed by Huynh [6]. Since our intention was to model the light projector and the camera independently the two steps calibration method was preferred.

## 2.1 Omnidirectional Camera Model

Assuming that the pair camera-mirror possesses a SVP, the omnidirectional camera can be modelled as the projection onto the sphere followed by the projection to a plane, as stated by Geyer and Daniilidis in [2]. Another way of approaching camera calibration is by considering the mirror surface as a known revolution shape and modelling it explicitly, for instance considering that the reflecting surface is a paraboloid and the camera is orthographic. Both models were tested and the comparative results were reported in [9]. The omni camera used for this work has a SVP but contains two reflecting surfaces so the first mentioned method was preferred.



**Fig. 2.** Image formation using the projective equivalence of a SVP catadioptric projection with the projection on the sphere.

The calibration is performed using a set of known 3D points distributed on the four walls of a cube placed around the sensor. Consider a scene point  $P_w = [x_w, y_w, z_w]$ , and  $P_s = [x_s, y_s, z_s]$  the intersection of the light ray emitted by the point  $P_w$  with the sphere of radius  $R = 2h$  (see Figure 2). We can write equation (1) where all points are represented with respect to the camera coordinate system.

$$\begin{cases} x_s = \lambda \cdot x_w \\ y_s = \lambda \cdot y_w \\ z_s = \lambda \cdot z_w \end{cases} \quad (1)$$

Since the points belong to the sphere:  $x_s^2 + y_s^2 + z_s^2 = R^2$ .

The perspective projection of  $P_s$  on the image plane from a point  $C = [0, \xi]$  produces a point  $P_i = [x, y]$  as expressed in equation (2)

$$\begin{cases} \frac{x_s}{\xi - z_s} = \frac{x}{\xi + \varphi} \\ \frac{y_s}{\xi - z_s} = \frac{y}{\xi + \varphi} \end{cases} \quad (2)$$



Adding the intrinsic camera parameters  $\alpha_u, \alpha_v, u_0, v_0$ , the pixel coordinates of the image points are shown in eq. (3)

$$\begin{cases} u = \frac{\alpha_u(\xi+\varphi)x_w}{\xi\sqrt{x_w^2+y_w^2+z_w^2-z_w}} + u_0 \\ v = \frac{\alpha_v(\xi+\varphi)y_w}{\xi\sqrt{x_w^2+y_w^2+z_w^2-z_w}} + v_0 \end{cases} \quad (3)$$

The parameters of the model are  $\xi$ , which depends on the eccentricity;  $\varphi$  which is a function of both the eccentricity and the scale;  $\alpha_u, \alpha_v, u_0, v_0$ , the intrinsic camera parameters;  $r_X(\phi), r_Y(\theta), r_Z(\varphi)$ , and  $t_x, t_y, t_z$ , the six extrinsic parameters that model respectively the orientation and the translation between the world coordinate system placed in the upper corner of the first calibration plane and the camera coordinate system. The orientation vectors are functions of the three angles  $(\phi, \theta, \varphi)$  which define the rotation on each axis and are expressed in radians while the translations are measured in millimeters, as detailed in [11].

The difference between the positions of the calculated image points and the positions of the real image points is the calibration error of the model. Minimizing the above error by means of an iterative algorithm such as Levenberg-Marquardt the model of the omnidirectional camera is calibrated.

## 2.2 Omnidirectional Laser Projector Model

The omnidirectional light projector is formed by a laser which emits a circle and is pointed to a conical mirror so that the projected light covers the entire field of view of the catadioptric camera. The proposed projector can be seen as a reversed omni-camera where the light flows in the opposite sense. So, the projector benefits of the attributes revealed by previous studies of catadioptric cameras based on the conical mirror shape. Lin and Bajcsy [7] pointed out that the conical mirror can be used for building true SVP configurations with the advantage that it preserves image points brightness better than other mirrors since it does not distort the image in longitudinal directions. Yagi [14] highlighted the fact that the conical mirror on vertical section behaves like a planar mirror and consequently provides a much better resolution than any other omni-mirror shape. Baker and Nayar [1] proved that the curved mirrors (such as parabolic, hyperbolic, etc.) increase defocus blur because of their bend. Consequently, the cone bears out to be the ideal shape of mirror to be used for building the structured light projector.

Unlike the camera, the light projector does not provide “image points” therefore no correspondences can be established. The bright spots on the scene are observed by the calibrated omnidirectional camera which possesses an unique center of projection. This property allows calculating the direction of the light source for each image point. Since the locations of the calibration planes are known, the 3D coordinates of the laser-stripe lying on those planes can be determined. A set of such points can be used for calibrating the pair laser-mirror.

Ideally, when the laser is perfectly aligned with the conical mirror, the 3D shape formed by the reflected laser pattern can be imagined as a circular cone, called “laser-cone”. Unfortunately, the precision of obtaining the coordinates of the bright spots is bounded by the catadioptric camera calibration accuracy and by its resolution. Moreover, a perfect alignment of the laser and the conical mirror is difficult to guarantee so a

more general shape than the circular cone should be considered. Since the perspective projection of a circle placed on a plane  $\Pi$  onto a plane that is not parallel with  $\Pi$  is an ellipse it can be deduced that a suitable shape for modelling the laser-cone is a revolution surface whose intersection with the plane perpendicular on the omnidirectional camera optical axis is an ellipse. This shape, the elliptic cone, was used in [9] and proves to be more accurate than the circular cone. Still, for a large amount of noise, the elliptical cone can not be uniquely determined.

Therefore, the general quadratic surface was chosen for modelling the laser projection. Consider  $P_{wi}(x, y, z)$  the bright spots on the calibration walls with known coordinates. The quadratic surface that passes through all the points is represented in eq. 4. Let  $H$  be the matrix that contains the coordinates of the points,  $A$  the matrix of the parameters and  $F$  the free term matrix. Writing  $H \cdot A = F$ , the matrix  $A$  can be obtained by  $A = (H' \cdot H)^{-1} \cdot H' \cdot F$ . This is a simple method for calibrating the omni projector. Since no iterations are needed it is much faster than the iterative minimization methods. However, its main drawback is that the matrix  $H$  can not be controlled and, for noisy data, it is likely to be singular.

$$\begin{bmatrix} x_1^2 & y_1^2 & z_1^2 & 2x_1y_1 & 2x_1z_1 & 2y_1z_1 & 2x_1 & 2y_1 & 2z_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i^2 & y_i^2 & z_i^2 & 2x_iy_i & 2x_iz_i & 2y_iz_i & 2x_i & 2y_i & 2z_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & y_n^2 & z_n^2 & 2x_ny_n & 2x_nz_n & 2y_nz_n & 2x_n & 2y_n & 2z_n \end{bmatrix} \cdot \begin{bmatrix} a_{11} \\ a_{22} \\ a_{33} \\ a_{12} \\ a_{13} \\ a_{23} \\ \beta_{31} \\ \beta_{32} \\ \beta_{33} \end{bmatrix} = \begin{bmatrix} \vdots \\ -1 \\ -1 \\ -1 \\ \vdots \end{bmatrix} \quad (4)$$

Therefore, a more robust method for finding the parameters of the general quadratic surface must be considered. Lets assume, without loss of generality, that the world reference system is placed such that the calibration planes are perpendicular on the  $X$  and  $Y$  axis. The intersections of the quadratic with the calibration planes are arcs described by a subinterval of the parameter domain: the arcs contained in the planes perpendicular on the  $X$  and  $Y$  axis provide information on the parameters of the quadratic with  $x = ct$  and  $y = ct$ , respectively. Writing the quadratic as in eq. 5, its intersection with the planes  $X$  and  $Y$  are shown in eq. 6 and eq. 7, respectively. The parameters of the arcs for each plane are obtained by fitting the corresponding points into the subsequent equations. Taking into account that the  $3 \times 3$  matrix is symmetric, the full set of parameters of the quadratic surface can be retrieved from equations 6 and 7.

$$[x \ y \ z] \cdot \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + [x \ y \ z] \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + f = 0 \quad (5)$$

$$[y \ z] \cdot \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} y \\ z \end{bmatrix} + [y \ z] \cdot \begin{bmatrix} P_x \\ Q_x \end{bmatrix} + R_x = 0 \quad (6)$$

$$[x \ z] \cdot \begin{bmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ z \end{bmatrix} + [x \ z] \cdot \begin{bmatrix} P_y \\ Q_y \end{bmatrix} + R_y = 0 \quad (7)$$

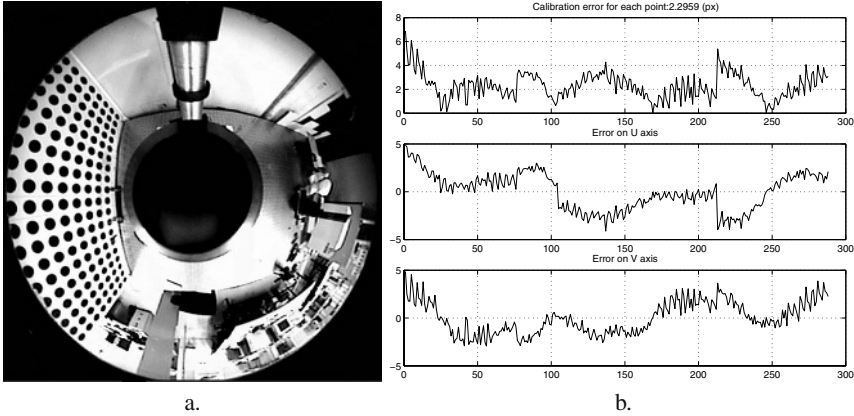


Fig. 3. a. One of the calibration planes. b. Error for the 285 calibration points, measured in pixels.

Table 1. The calibrated parameters for the omni camera.

$\xi$	$\varphi$	$\alpha_u$	$\alpha_v$	$u_0$	$v_0$	$r_x$	$r_y$	$r_z$	$t_x$	$t_y$	$t_z$
1.06	-9.64	-32.53	33.24	429.51	292.72	0.02	0.01	-0.009	-26.45	-0.82	-754.1

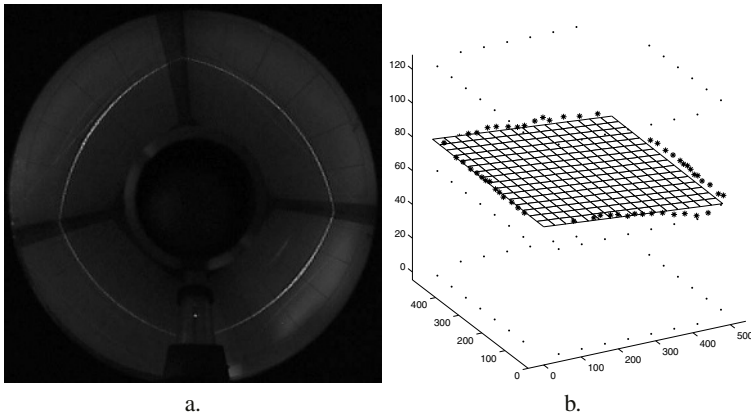
Dividing the calibration in two parts the number parameters to be simultaneously minimized decreases which leads to a robust calibration method.

### 3 Experimental Results

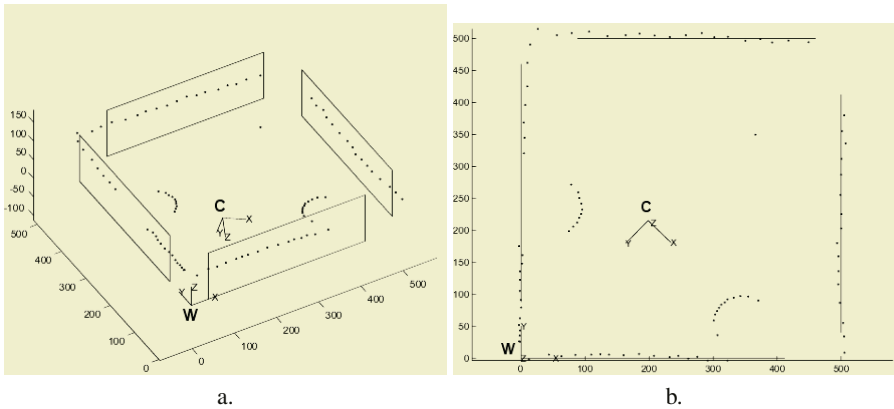
The system was built using off the shelf components. The optics and the mirror used for the omnidirectional camera were provided by Remote Reality [10]. The camera is a Sony SSC-DC198P with the ccd of 1/3". The laser and its optics are produced by Lasiris, the diode power is 3mW and produces red light with a wavelength of 635nm.

The camera calibration is performed using a set of 285 dots distributed on the four planes placed around the sensor. The distance between the centers of any two adjacent dots on the same plane is 6cm and the height of the calibration plane is 80cm. A semi-automatic point extraction method is performed. For each plane, several dots are selected by the user and their centers are determined with sub-pixel accuracy. The centers of the remaining dots are automatically found with the same precision. The calibrated parameters of the camera-model are listed in the Table 1. The average calibration error is  $\mu = 2.3\text{px}$  and the sample standard deviation  $\sigma = 2.542$ .

The conical mirror used for building the laboratory prototype has a height  $h = 4.4$  cm and the cone aperture angle is  $\beta = 52$  degrees. The laser projects a circular cone with a fan angle  $\alpha = 11.4$  degrees. Given that the relation between the two angles is  $\beta \approx 0.5(\alpha + \pi/2)$ , the laser is reflected along a very flat surface which can be approximated to a plane:  $ax + by + cz + d = 0$ , see Figure 4.b. The center of the laser stripe is determined with sub-pixel accuracy using the peak detection method described by Forest [4] and the discrete points are used for calibrating the parameters of the plane:  $a = -0.13$ ,  $b = -0.001$ ,  $c = 1$  and  $d = 78.99$ .



**Fig. 4.** a. Projection of the laser pattern. b. Flat surface fitted to a set of discrete points from the laser stripe. The three dotted rectangles are the points used for calibrating the camera.



**Fig. 5.** Omnidirectional 3D profile obtained along the laser stripe. The dots stand for the reconstructed 3D points. a. Lateral view b. Upper view.

With the sensor surrounded by four planes depth was calculated using a set of discrete points of the laser pattern. For a scene containing two cylinders the result is presented in Figures 5 with the two cylindrical shapes correctly identified. It is also noticeable that the points on the walls fall on the corresponding planes. In terms of accuracy, the radius of the cylinder was measured and has 93cm while the range finder returned a result of 95cm.

## 4 Conclusions

It is noticeable that the use of 360 degrees images and of scene-depth information is ideal for robot navigation tasks. Starting from this observation we combine the advantages of omnidirectional vision and structured light. We presented here the geometry

and the calibration for a prototype of a panoramic range finder. The two omnidirectional systems that compose the sensor are calibrated and the resulting model is used for measuring depth in a real scene. The accuracy of the sensor is enhanced by the use of sub-pixel accuracy techniques at calibration and reconstruction stages. The results obtained are encouraging and prove that this sensor can be used in real robot navigation and depth perception applications.

## References

1. S. Baker and S.K. Nayar. A theory of catadioptric image formation. *IEEE Int. Conf. on Computer Vision, ICCV*, pages 35–42, 1998.
2. K. Daniilidis C. Geyer. A unifying theory for central panoramic systems and practical applications. *Sixth European Conference on Computer Vision*, pages 445–461, June 2000.
3. M. Fiala and A. Basu. Feature extraction and calibration for stereo reconstruction using non-svp optics in a panoramic stereo-vision sensor. In *Workshop on Omnidirectional Vision*, pages 79–86, 2002.
4. J. Forest, J. Salvi, E. Cabruja, and C. Pous. Laser stripe peak detector for 3d scanners. a fir filter approach. In *International Conference on Pattern Recognition*, volume 3, pages 646–649, Cambridge, United Kingdom, August 2004.
5. J. Gluckman and S.K. Nayar. Planar catadioptric stereo: Geometry and calibration. *IEEE Computer Vision and Pattern Recognition*, 1(1):I: 22–28, 23-25 June 1999.
6. Du Q. Huynh. Calibration of a structured light system: a projective approach. In *IEEE Computer Vision and Pattern Recognition*, pages 225–230, 17-19 June 1997.
7. S. S. Lin and R. Bajcsy. The true single view point (svp) configuration for omni-directional view catadioptric system using cone mirror. Technical report ms-cis-00-24, Computer and Information Science Department, University of Pennsylvania., Philadelphia, PA, USA, 11 2001.
8. Radu Orghidan, Joaquim Salvi, and El Mustapha Mouaddib. Calibration of a structured light-based stereo catadioptric sensor. *Workshop on Omnidirectional Vision*, IEEE Conf. on Computer Vision and Pattern Recognition - Volume 7, 2003.
9. Radu Orghidan, Joaquim Salvi, and El Mustapha Mouaddib. Omnidirectional depth computation from a single image. *IEEE International Conference on Robotics and Automation, ICRA*, April 18-22, 2005.
10. Remote Reality. <http://www.remotereality.com/>.
11. J. Salvi, X. Armanague, and J. Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7):1617–1635, July 2002.
12. J. Salvi, J. Batlle, and E. Mouaddib. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, (19):1055–1065, September 1998.
13. Tomas Pajdla Tomas Svoboda. Panoramic cameras for 3d computation. *Proceedings of the Czech Pattern Recognition Workshop*, pages 63–70, February 2000.
14. Y. Yagi. Omnidirectional sensing and its applications. *IEICE Trans on Information and Systems*, E82-D(3):568–578, 1999.

# Probabilistic Object Tracking Based on Machine Learning and Importance Sampling

Peihua Li<sup>1</sup> and Haijing Wang<sup>2</sup>

<sup>1</sup> College of Computer Science and Technology, Heilongjiang University  
Heilongjiang Province, 150086, China  
peihualj@hotmail.com

<sup>2</sup> School of Computer Science and Technology, Harbin Institute of Technology, China

**Abstract.** The paper presents a novel particle filtering framework for visual object tracking. One of the contributions is the development of a likelihood function based on one of machine learning algorithm—AdaBoost algorithm. The likelihood function can capture the structure characteristics of one class of objects, and is thus robust to clutters and noise in the complex background. The other contribution is the adoption of mean shift iteration as a proposal distribution, which can steer discrete samples towards regions which most likely contain the targets, and is therefore leading to computational efficiency in the algorithm. The effectiveness of such a framework is demonstrated with a particular class of objects—human faces.

## 1 Introduction

Particle filtering is widely investigated in recent years in computer vision, because of its powerful ability to deal with general non-linear and non-Gaussian problems. Particularly in visual tracking, measurement model (likelihood function) is often non-linear due to clutter or noise in the background [1], causing the posterior distribution of the system state being non-linear. It is why particle filtering receives so much attention in the domain. Two factors weight heavily for the effectiveness of particle filter. One is likelihood function, responsible for extracting visual information from images. The other is the proposal distribution, from which a set of discrete samples will be drawn. The paper has contributions in both aspects.

### 1.1 Likelihood in Particle Filter

Many researcher are devoted to development of a effective measurement likelihood. Isard et al. presents a contour likelihood function based on edges [1]. The measurement is performed along the normal lines to the discrete sampling points on the contour, and the Canny edge detector is applied to these normals to obtain the local maximum as features. Under the assumption of the feature outputs on distinct normal lines are statistically independent, together with some other assumptions, a likelihood function is derived and used in the

framework of particle filter. Chen et al. [2], who argue that the measurement of adjacent normals is statistically dependent, extends the above likelihood in the framework of Hidden Markov Model (HMM), integrating the edge as well as color information. Nummiaro et al. [3] adopted a metric defined on weighted multi-channel color histogram [4], which represents the target distribution, as the likelihood in the framework of particle filter [3].

We have also seen that machine learning has gradually played an important role in the design of visual measurement model. Mikolańczyk et al. [5] incorporates the face detector of Schneiderman and Kanade [6] into particle filter for face detection and tracking. Two detectors –frontal and profile face detectors are combined to estimate the pose and give measurement probability. Avidan integrates the Support Vector Machine (SVM) classifier into an optical flow, and maximizes the SVM classification score, instead of minimizing the intensity difference function between successive frames [7]. Furthermore, an approach of Gaussian pyramid in both learning and tracking stages is introduced to handle large motions in image plane.

Motivated by these work, we independently propose a likelihood function based on AdaBoost algorithm. This likelihood function provides the probability of a measurement given the input image, in addition, the computation of which is efficient, as will be explained in the next section. This kind of likelihood is particularly suited to some classes objects, e.g. faces, cars and pedestrians tracking.

## 1.2 Proposal Distribution in Particle Filter

How to get an effective and efficient proposal distribution is a challenging problem. Isard et al. [8] proposes an importance-sampling method, which relies on an independent global segmentation and tracking of human-skin block. Li et al. [9] introduces proposal distribution based on Kalman filter and unscented Kalman filter, which depends on the learned motion model and edge-based likelihood. Wu et al. [10] present a novel particle filter, as an approximation of a factorized graphical model, in which shape and color samples are interactively drawn from each other's measurements based on importance sampling.

In the paper, we introduce a general and efficient proposal distribution (importance function) resulting from mean shift iteration. It is general because it use weighted multiple-channel color histogram to represent the distribution of the object, not specific to, for instance, human skin color; it is efficient because the optimization of the metric based on gradient descent is fast which measures the similarity of two distributions defined on the target and candidate [4].

## 2 Likelihood Function Based on AdaBoost Algorithm

In the face detection area, Viola and Jones [11] first realize the selection of critical visual features from a large set of Harr-like features and the training of

Adaboost simultaneously [12]. Thanks to the introduction of a new image representation called “Integral Image”, which allows the features used to be computed efficiently, and combination of weak classifiers in a cascade, which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions, their algorithm is computationally efficient.

Based on the work of Viola and Jones and the Real AdaBoost algorithm [13] that can give conditional probability density of an test belonging to one specific class, as well as the potential of machine learning in visual tracking, the authors propose training a likelihood function. The main idea is that since in probabilistic visual tracking, we are concerned with the probability of a candidate (expressed by a system state), we therefore look forward to training such an likelihood function which captures the structural characteristics of one class of objects and gives a probabilistic interpretation. There is, however, a fundamental difference for the use of AdaBoost in the paper from that in [11]. For the purpose of face detection (classification problem—two classes: face or nonface) in one image, they perform exhaustive search at different locations and at different possible scales. They thus adopt cascade structure to reject gradually candidate regions that most probably contain non-faces. Whereas in tracking, we are only concerned with one image candidate region and its probability belonging to the target, so cascade structure is not necessary any more.

## 2.1 Training a Likelihood Function Based on AdaBoost Algorithm

To accommodate face poses variations, we collect training examples which includes faces in different views: frontal, half-left and right profiles, left and right profiles, and in each views, the faces demonstrate a degree of upside-down rotations and in-plane rotation. These training examples are collected widespread in Internet as well as captured in our lab. Some of non-face examples are collected in internet, others are randomly sampled from windows in the image dataset.

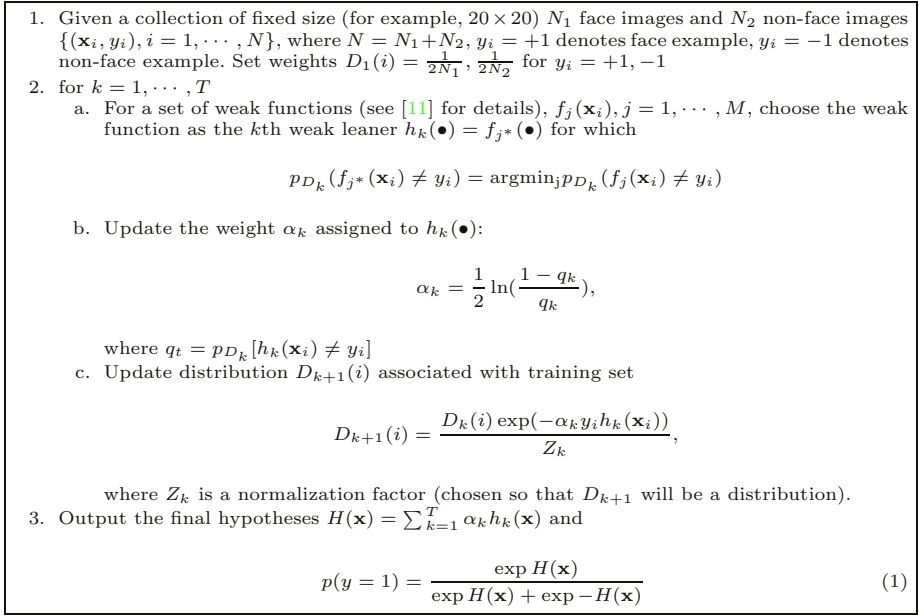
The training of the likelihood function is illustrated in Figure 1:

The output Equ. (1) of the real AdaBoost algorithm has a probabilistic interpretation, which gives a probability of an image patch  $\mathbf{x}$  belonging to human faces. The justification of Equ. (1) may be found in [13].

## 3 A Particle Filtering Framework for Object Tracking

In the paradigm of particle filtering (also known as sequential importance sampling) [9], the system is described by  $p(\mathbf{X}_k|\mathbf{X}_{k-1})$ ,  $p(\mathbf{Y}_k|\mathbf{X}_k)$ . The transition prior  $p(\mathbf{X}_k|\mathbf{X}_{k-1})$  indicates the the evolution of the state is a Markov process, and  $p(\mathbf{Y}_k|\mathbf{X}_k)$  denotes the observation density (likelihood function) in the dynamical system, in which the measurements are conditionally independent of each other given the states. The posterior density is approximated by a set of discrete samples, called particles,  $\{(\mathbf{X}_k^{(i)}, \omega_k^{(i)}, i = 1 \dots, N\}$ . The computation





**Fig. 1.** The AdaBoost algorithm for training face detector.

of weights concerns an introduction of an importance function, called a proposal density,  $\pi(\mathbf{X}_k | \mathbf{X}_{k-1}, \mathbf{Y}_{1:k})$ , from which particles can be easily drawn, and which approximates the posterior density. As such, the weights can be computed iteratively as follows

$$\pi(\mathbf{X}_k | \mathbf{X}_{k-1}, \mathbf{Y}_{1:k}) = \frac{p(\mathbf{Y}_k | \mathbf{X}_k^{(i)}) p(\mathbf{X}_k^{(i)} | \mathbf{X}_{k-1}^{(i)})}{\pi(\mathbf{X}_k | \mathbf{X}_{k-1}^{(i)}, \mathbf{Y}_{1:k})} \quad (2)$$

### 3.1 Mean Shift Iteration

Mean shift iteration is targeted at seeking the candidate which has the most similar distribution with the target in a local region [4]. The search is based on gradient optimization of a scale-invariant metric between target and candidate distribution

$$d(q, \tilde{p}(\mathbf{y})) = \sqrt{1 - \rho(q, \tilde{p}(\mathbf{y}))} \quad (3)$$

where  $\rho(q, \tilde{p}(\mathbf{y}))$  is Bhattacharyya coefficient. The distribution is generally in the form of weighted multi-channel color histogram,  $q = \{q_u\}_{u=1, \dots, m}$  with  $\sum_{u=1}^m q_u = 1$  for target, and  $\tilde{p}(\mathbf{y}) = \{p_u(\mathbf{y})\}_{u=1, \dots, m}$  with  $\sum_{u=1}^m p_u = 1$  for candidate. In this case,  $\rho(q, \tilde{p}(\mathbf{y})) = \sum_{u=1}^m \sqrt{p_u(\mathbf{y}) q_u}$ . Let us denote  $\mathbf{z}_i$   $i = 1, \dots, n$  the pixel locations of one face candidate, centered at  $\mathbf{y}$  in the current frame, the distribution of the face candidate can be expressed as  $\tilde{p}(\mathbf{y}) = \{p_u(\mathbf{y})\}_{u=1, \dots, m}$ , where

1. Initialisation  
Draw particles from the prior  $p(\mathbf{X}_0^{(i)})$  to obtain a set  $\{(\tilde{\mathbf{X}}_0^{(i)}, 1/N), i = 1, \dots, N\}$ ,
2. Sampling and updating step  
for  $i = 1, \dots, N$  :
  - a. Generate a random number  $\alpha \in [0, 1]$ , uniformly distribution.
  - b. If  $\alpha < q$  use mean shift algorithm to determine proposal distribution. Specifically, use mean shift algorithm in the current frame to seek the state  $\hat{\mathbf{X}}_k$ , which has the most similar distribution with  $\tilde{\mathbf{X}}_{k-1}$ . Draw  $\mathbf{X}_k^{(i)}$  from  $\mathcal{N}(\hat{\mathbf{X}}_k, \mathbf{P})$ . Compute the proposal distribution  $\pi(\mathbf{X}_k^{(i)} | \tilde{\mathbf{X}}_{k-1}^{(i)}, \mathbf{Y}_{1:k})$  according to
 
$$\pi(\mathbf{X}_k^{(i)} | \tilde{\mathbf{X}}_{k-1}^{(i)}, \mathbf{Y}_{1:k}) = \sqrt{1 - \rho(q(\tilde{\mathbf{X}}_{k-1}), \tilde{p}(\mathbf{X}_k^{(i)}))}$$
 and then compute the weight of the sample  $\mathbf{X}_k^{(i)}$  according to Equ. (2)
  - c. If  $\alpha \geq q$  use the transition prior  $p(\mathbf{X}_k | \mathbf{X}_{k-1})$  as the proposal distribution. Draw  $\mathbf{X}_k^{(i)}$  from the proposal distribution. Compute the weight of the sample  $\mathbf{X}_k^{(i)}$ 

$$\tilde{\omega}_k^{(i)} = p(\mathbf{Y}_k | \mathbf{X}_k^{(i)})$$
3. Output step  
Output a set  $\{(\mathbf{X}_k^{(i)}, \omega_k^{(i)}), i = 1, \dots, N\}$  of particles that can be used to approximate the posterior distribution as  $p(\mathbf{X}_k | \mathbf{Y}_{1:k}) \approx \sum_{i=1}^N \omega_k^{(i)} \delta(\mathbf{X}_k - \mathbf{X}_k^{(i)})$ , and the system mean (tracking result)  $\hat{\mathbf{X}}_k \approx \sum_{i=1}^N \omega_k^{(i)} \mathbf{X}_k^{(i)}$
4. Selection (resampling) step  
Resample the particles  $\{(\mathbf{X}_k^{(i)}, \omega_k^{(i)})\}$  with probability  $\omega_k^{(i)}$  to obtain  $N$  i.i.d random particles  $\{\tilde{\mathbf{X}}_k^{(i)}, 1/N\}$ , approximately distributed according to  $p(\mathbf{X}_k | \mathbf{Y}_{1:k})$
5.  $k = k + 1$ , go to step 2.

**Fig. 2.** The framework for visual tracking.

$$p_u(\mathbf{y}) = \frac{1}{\sum_{i=1}^n k(\|\frac{\mathbf{y} - \mathbf{z}_i}{h}\|^2)} \sum_{i=1}^n k(\|\frac{\mathbf{y} - \mathbf{z}_i}{h}\|^2) \delta(b(\mathbf{z}_i) - u) \quad (4)$$

where  $h$  is the radius of a candidate region,  $b(\mathbf{z}_i)$  is a function which associates to the pixel at location  $\mathbf{z}_i$  the index  $b(\mathbf{z}_i)$  of the histogram, and  $\delta(\cdot)$  is the Kronecker delta function. The weighting function is adopted as Epanechnikov kernel.

The distribution of target is adopted as that of the tracking result in the previous frame and has similar form to Equ. (4).

### 3.2 Tracking Algorithm

Apart from the initialization, the framework operates in three steps: sampling and updating, output, and re-sampling (or selection) step. While the mean shift is efficient [4] in seeking the promising target, it depends on color information and is thus not robust to lighting changes [3]. So we will not draw all particles from the proposal distribution (step 2b): some will be sampled from the transition prior for diversity of particles (step 2c). Let  $\hat{\mathbf{X}}_k$  be the converged result of mean shift iteration, it is reasonable and simple to assume that the distribution of the potential target state is Gaussian  $\mathcal{N}(\hat{\mathbf{X}}_k, \mathbf{P})$ , where  $\hat{\mathbf{X}}_k$  is the mean and  $\mathbf{P}$  is the covariance. The detailed algorithm is presented in Figure 2.

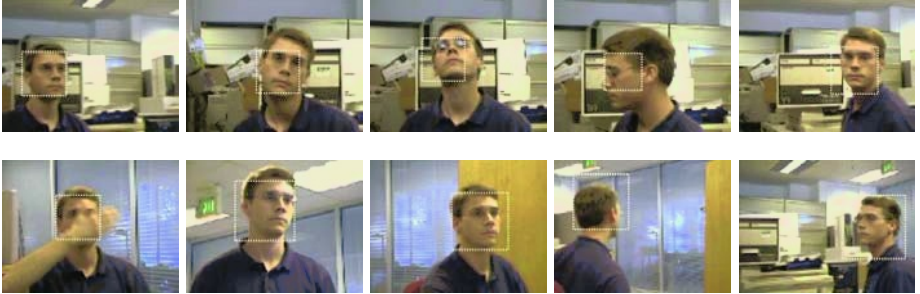
## 4 Experiments

The initialization of the particle filtering algorithm is accomplished using AdaBoost face detector of Viola and Jones [11]. The transition prior is a random walk

$$\mathbf{X}_k = \mathbf{X}_{k-1} + \mathbf{Q}_k \quad (5)$$

where  $\mathbf{X}_k = [x_k, y_k, s_k]$ ,  $(x_k, y_k)$  are the coordinate of the center of the tracked region and  $s_k$  is the scale. The likelihood function is represented by Equ. (1).

The algorithm is implemented with Visual C++ 5.0 on a laptop of Pentium IV-2.2GHz CPU with Microsoft XP. In the image sequence, both the camera and the subject are moving, and the motion of target is agile and large. The background is complex and in some snapshots the color resembles to human face. The algorithms rely only on edge [1], or rely only on relying only on color [4], fail to track object. Our likelihood is aimed at finding out the structural information of human face and is able to neglect background clutter. Together with the aid of mean shift as importance sampling function, and the proposed algorithm can robustly track the face in real time (about 20 ms) throughout the whole image sequence. Some of typical tracking result are demonstrated in Figure 3.



**Fig. 3.** Some of tracking results in the image sequence. It can be seen that complex background, significant pose variations, partial occlusion are all well dealt with.

## 5 Conclusions and Discussion

In the paper, a novel likelihood function is developed based on AdaBoost training algorithm, which is capable of capturing the structural characteristics of the human faces and gives a probability interpretation, and is not sensitive to illumination changes. Furthermore, the general and efficient mean shift iteration is considered as a means to produce the proposal distribution in the particle filter, which can steer the particles towards most probable locations of target in images and thus leads to efficacy of the algorithm. Although in the area of face detection, a single classifier trained on all poses appears to be inaccurate, the likelihood function, trained on all poses based on probabilistic version of

AdaBoost, works satisfactorily in tracking. Experiments show that the particle filter, with the proposed likelihood function but without mean shift iteration as the proposal distribution, can well track the object, yet with the longer tracking time. It is also found that mean shift iteration plays a much larger part when the features of faces are not distinct, for example, in poses beyond profiles. The validation of the framework is demonstrated with experiments dealing with tracking of human faces. However, it can naturally extend to some other categories of objects, for example, pedestrians and cars. Future research will focus on this aspect.

## References

1. Isard, M., Blake, A.: *Cotour Tracking By Stochastic Propagation of Conditional Density*. European Conf. Comp. Vis. Cambridge UK (1996) 343–356
2. Chen, Y., Rui, Y., Huang, T.S.: *JPDFAF Based HMM for Real-Time Contour Tracking*. IEEE Int. Conf. on Comp. Vis. and Pat. Rec. Hawaii USA(2001) 543–550
3. Nummiaro, K., Koller-Meier, E., Gool, L.V.: *An Adaptive Color-Based Particle Filter*. Image and Vision Computing **21** (2003) 100–110
4. Comaniciu, D., Ramesh, V., Meer, P.: *Real-time Tracking of Non-rigid Objects Using Mean Shift*. IEEE Int. Conf. on Comp. Vis. and Pat. Rec. South Carolina USA(2000) 142–149
5. Mikolajczyk, K., Choudhury, R., Schmid, C.: *Face Detection in a Video Sequence – a Temporal Approach*. IEEE Int. Conf. on Comp. Vis. and Pat. Rec. Hawaii USA(2001) 96–101
6. Schneiderman, H., Kanade, T.: *A Statistical Method for 3D Object Detection Applied to Faces and Cars*. IEEE Int. Conf. on Comp. Vis. and Pat. Rec. South Carolina USA(2000) 746–751
7. Avidan, S.: *Support vector tracking*, IEEE Int. Conf. on Comp. Vis. and Pat. Rec. Hawaii USA(2001) 184–191
8. Isard, M., Blake, A.: *ICondensation: Unifying Low-level and High-level Tracking in a Stochastic Framework*. European Conf. Comp. Vis. Freiburg Germany (1998) 893–908
9. Li, P., Zhang, T., Pece, A.E.C.: *Visual Contour Tracking based on Particle Filters*. Image and Vision Computing **21** (2003) 111–123
10. Wu, Y., Huang, T.S.: *A Co-inference Approach to Robust Visual Tracking*. IEEE Int'l Conf. Comp. Vis. Vancouver Canada (2001) 26–33
11. Viola, P., Jones, M.J.: *Robust Real-time Object Detection*. IEEE Workshop on Statistical and Computational Theories of Vision. Vancouver Canada (2001)
12. Freund, Y., Schapire, R.E.: *A Decision-theoretic Generalization of On-line Learning and An Application to Boosting*. Journal of Computer and System Sciences **55**(1) (1997) 119–139
13. Friedman, J., Hastie, T., Tibshirani, R.: *Additive Logistic Regression: a Statistical View of Boosting*. Annals of Statistics. **28** (2000) 337–374

# A Calibration Algorithm for POX-Slits Camera

Nuno Martins<sup>1</sup> and Hélder Araújo<sup>2</sup>

<sup>1</sup> DEIS, ISEC, Polytechnic Institute of Coimbra, Portugal

<sup>2</sup> ISR/DEEC, University of Coimbra, Portugal

**Abstract.** Recent developments have suggested alternative multiperspective camera models potentially advantageous for the analysis of the scene structure. Two-slit cameras are one such case. These cameras collect all rays passing through two lines. The projection model for these cameras is non-linear, and in this model every 3D point is projected by a line that passes through that point and intersects two slits. In this paper we propose a robust non-iterative linear method for the calibration of this type of cameras. For that purpose a calibrating object with known dimensions is required. A solution for the calibration can be obtained using at least thirteen world to image correspondences. To achieve a higher level of accuracy data normalization and a non-linear technique based on the maximum likelihood criterion can be used to refine the estimated solution.

## 1 Introduction

Projection models constitute a relevant issue in computer vision. The mathematical model that describes the formation of the most common type of images is the perspective projection model. Most of the commercialized optical devices generate images whose geometrical properties are described in this model. Therefore, the classic pinhole and orthographic camera models have long been used in 3D imaging applications.

However certain special vision problems can benefit from the application of alternative projection models, as recent developments have suggested. Besides, those developments in image sensing make the perspective model highly restrictive. These multiperspective models have been providing advantageous imaging systems for understanding the structure of observed 3D scenes. Examples of such camera models are bi-centric [13], crossed-slits (also known as x-slits) [15], general linear [14] and rational polynomial [4] models. Multiperspective imaging has also been explored in computer graphics[8].

In the bi-centric model the centers of horizontal and vertical projections lie in different locations on the camera's optical axis. Perspective and pushbroom cameras [3] are particular cases of this model, if the horizontal and vertical projections lie in the same locations and if only the horizontal projection resides on the infinity (corresponds to a vertical strip of a sensor translating sideways), respectively. In [13] it was also shown that a straight line in the scene is projected into a hyperbole in the image. The pushbroom model collects rays along parallel

planes from points swept along a linear trajectory [3]. The most visible distortion in the images that follow the pushbroom model is the variation of aspect-ratio.

General linear and cubic camera are general models. The general linear camera model unifies most projection models used in computer vision, including perspective and affine models, optical distortions models, x-slits models, stereo systems and catadioptric systems [14]. A cubic camera maps the image points as rational polynomial functions, of degree less than four, of the coordinates of a world point [4]. This camera model treats projective, affine, pushbroom and x-slits cameras as particular cases.

In the x-slits model the projection ray of a generic 3D point is defined by the 3D line that passes through the point and two lines, referred as slits. The image is obtained by the intersection of every projective ray with the image plane. This model was initially designed by one of the pioneers of the color photography, Ducos du Hauron, in 1888 [7], under the name “*transformisme en photographie*” [6]. He thought that his device would be used in the 20<sup>th</sup> century to “*create visions of another world*” [7]. However, it was a restricted model in terms of the slits positions, which were parallel and orthogonal between each other (this situation was later referred as parallel-orthogonal x-slits, or pox-slits [15]). An interesting aspect is that pox-slits projection equations are similar to the bi-centric model [1]. A particular case of the pox-slits camera, in which the vertical slit resides at infinity, is the pushbroom camera.

One century later, the pox-slits model was revised and generalized by Kingslake, who concluded that it was similar to the perspective projection model in which the image is stretched or compressed in one direction more than the other [6]. This fact shows its adequacy to the use in wide-screen technologies.

Zomet *et al*, in [15], expanding the Kingslake generalization, introduced the x-slits projection model. According to their study, one advantage of of this model is the fact that x-slits images can be easily generated by perspective images. Shortly, this procedure is performed by pasting together vertical or horizontal samplings of a sequence of images captured from a perspective camera, which moves, respectively, along a horizontal or vertical line. With a more complex procedure new x-slits views can be generated even when the camera motion is not parallel to the image plane [1]. The idea of sampling columns from images has been explored before, but using a constant sampling function [10]. This traditional mosaicing technique is similar to the one used to create pushbroom panoramas [13]. Another remarkable aspect of this camera is that perspective model is a particular case of the x-slits camera, in which the vertical and horizontal slits lie in the same plane. The optical center of the perspective camera is the intersection of the slits.

In spite of the extensive analysis of x-slits cameras by [15], they have only focused on aspects related to image generation. In this paper we deal with the problem of calibrating this type of cameras.

Grossberg *et al*, in [2], presented a different camera calibration algorithm, referred to as the generic imaging model. In that case calibration consists in determining, for every image pixel, the associated 3D projection ray. This method

is also used in [8] and [10]. This mapping can be conveniently described using a set of virtual sensing elements, called raxels. Raxels include geometric, radiometric and optical properties.

As an extension to [2], [12] introduce a generic calibration approach. In this method at least three images of a calibration object are acquired. The fact that a projective ray is a 3D line yields a constraint that allows the recovery of both the motion and the camera's parameters. This constraint is formulated via a set of trifocal tensors that can be estimated linearly. In [9] this calibration method is used in a 3D reconstruction process, with a parametric reprojection to refine the obtained solution, based on bundle adjustment.

In the calibration method described in this paper, we use the non-linear x-slits equations. For estimation purposes the equations are rewritten so that linear estimation methods can be used. For good levels of accuracy in the estimates, data normalization and a non-linear technique based on the maximum likelihood criterion can be used [5].

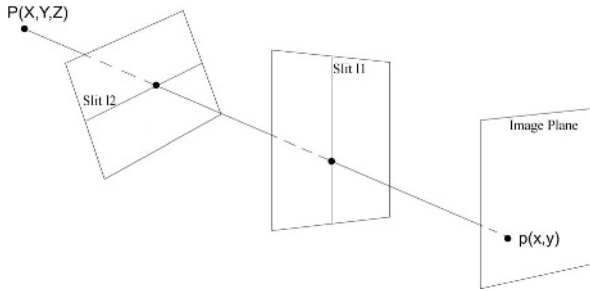


Fig. 1. X-slits projection model.

## 2 X-Slits Projection Model

Consider the x-slits projection configuration represented in figure 1. The projective ray of a generic 3D point,  $P$ , must intersect two lines, or slits,  $l_1$  and  $l_2$ . Point  $P$  together with each slit defines one plane. The intersection of those planes defines the projective ray. The projection of the 3D point in the image,  $p$ , is obtained by the intersection of the projective ray with the image plane.

To define the two slits, let  $u_i$  and  $v_i$  (with  $i = 1, 2$ ) be two generic planes defined in a space of 3 dimensions, given by their parametric coordinates. The slits,  $l_i$ , are defined through the intersection of those planes. These slits can be represented by the dual Plucker matrix [5], whose equation is

$$L_i^* = u_i v_i^T - v_i u_i^T = \begin{bmatrix} 0 & L_{i34} & L_{i42} & L_{i23} \\ -L_{i34} & 0 & L_{i14} & -L_{i13} \\ -L_{i42} & -L_{i14} & 0 & L_{i12} \\ -L_{i23} & L_{i13} & -L_{i12} & 0 \end{bmatrix}$$

if we use the Plucker coordinates of the slits.

The projective ray,  $l$ , is the intersection between two planes, defined by each slit  $l_i$  and the 3D point  $P$  ( $L_1^*P$  and  $L_2^*P$ , respectively), and can be defined by the dual Plucker matrix, through

$$L^* = (L_1^*P)(L_2^*P)^T - (L_2^*P)(L_1^*P)^T$$

Assuming that the image plane,  $I$ , is defined by the points  $P_0$ ,  $P_1$  and  $P_2$ , any point that belongs to  $I$  can be expressed by the linear combination of those points, given by

$$kxP_0 + kyP_1 + kP_2$$

As a result any point from a 2D space vector defined in the image plane, in homogeneous co-ordinates, is given by  $p = [kx \ ky \ k]^T$  [11].

The projection of a 3D generic point  $P$  in the image plane  $I$  generates a 2D point  $p$ . This projection is given by the intersection of the projective ray  $l$  with the image plane  $I$ . Therefore  $l$  must belong to both planes  $L_i^*P$ . Therefore,

$$\begin{bmatrix} P^T L_1^* P_0 & P^T L_1^* P_1 & P^T L_1^* P_2 \\ P^T L_2^* P_0 & P^T L_2^* P_1 & P^T L_2^* P_2 \end{bmatrix} p = 0 \quad (1)$$

The solution for equation (1) is the right null space of the matrix. This solution is obtained by using the cross product between the elements of the matrix, as

$$kp = \begin{bmatrix} P^T L_1^* (P_1 P_2^T - P_2 P_1^T) L_2^* P \\ P^T L_1^* (P_2 P_0^T - P_0 P_2^T) L_2^* P \\ P^T L_1^* (P_0 P_1^T - P_1 P_0^T) L_2^* P \end{bmatrix}$$

The homogeneous relation between 3D world scene points and 2D image points, in pixels, for the x-slits projection model is

$$kp = \begin{bmatrix} k_x & \gamma & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P^T L_1 I_0 L_2 P \\ P^T L_1 I_1 L_2 P \\ P^T L_1 I_2 L_2 P \end{bmatrix} \quad (2)$$

where  $I_0$ ,  $I_1$  and  $I_2$  are the Plucker matrices corresponding to the  $x$  and  $y$  axis of the image plane and the line at infinity.  $k_x$  and  $k_y$  are the focal lengths.  $(c_x, c_y)$  are the coordinates of the principal point and  $\gamma$  is the image skew. According to [15], this solution is unique unless it resides on the line joining the intersections of the two slits with the image plane.

### 3 Calibrating X-Slits Projection Model

In this section we describe an algorithm to calibrate the x-slits camera. We begin with a particular case of this camera, known as pox-slits, and then we address the general case.



### 3.1 Pox-Slits Case

We show how to calibrate the pox-slits camera because this camera is similar to the bi-centric camera and a generalization of the pushbroom camera.

Let us define,

$$u_1 = [1\ 0\ 0\ 0]^T \quad v_1 = [0\ 0\ 1\ -Z_1]^T \quad u_2 = [0\ 1\ 0\ 0]^T \quad v_2 = [0\ 0\ 1\ -Z_2]^T$$

Let us also consider three homogeneous points

$$P_0 = [1\ 0\ 0\ 0]^T \quad P_1 = [0\ 1\ 0\ 0]^T \quad P_2 = [0\ 0\ 0\ 1]^T$$

which belong to the image plane. As a result, equation (2) is given by

$$k \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & \gamma & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -Z_1 \frac{X}{Z-Z_1} \\ -Z_2 \frac{Y}{Z-Z_2} \\ 1 \end{bmatrix} \quad (3)$$

The calibration algorithm aims at estimating the intrinsic camera parameters  $k_x$ ,  $k_y$ ,  $c_x$ ,  $c_y$  and  $\gamma$  and the slits parameters  $Z_1$  and  $Z_2$ . From equation (3) we can obtain

$$\begin{aligned} -k_x Z_1 X Z + k_x Z_1 Z_2 X - \gamma Z_2 Y Z + \gamma Z_1 Z_2 Y + c_x Z^2 - c_x Z_1 Z - c_x Z_2 Z + \\ + c_x Z_1 Z_2 + Z_1 x Z + Z_2 x Z - Z_1 Z_2 x = x Z^2 \end{aligned} \quad (4)$$

and

$$c_y Z - c_y Z_2 - k_y Z_2 Y + Z_2 y = Z y \quad (5)$$

Assuming, without loss of generality,  $C_1 = c_y Z_2$  and  $C_2 = k_y Z_2$ , we can rewrite equation (5), matrix form, as

$$[Z \ -1 \ -Y \ y] \begin{bmatrix} c_y \\ C_1 \\ C_2 \\ Z_2 \end{bmatrix} = Z y$$

Using, at least, four world to image correspondences, we obtain a system of equations whose solution can be obtained using any numerical linear method, e.g, SVD. The solutions of this system of equations yield estimates for the intrinsic parameters  $k_y$  and  $c_y$ , and the slit parameter  $Z_2$ .

Assuming now, without loss of generality,  $C_3 = k_x Z_1$ ,  $C_4 = c_x Z_1$ ,  $C_5 = Z_1 \gamma$  and  $C_6 = c_x Z_1$ , and substituting the estimated parameters in equation (4) we get

$$\begin{aligned} x Z^2 - x Z Z_2 = (-X Z + X Z_2) C_3 - Y Z Z_2 \gamma + Y Z_2 C_5 + (Z^2 - Z Z_2) c_x + \\ + (-Z + Z_2) C_6 + (x Z - x Z_2) Z_1 \end{aligned}$$

Similarly, and using at least six world to image correspondences, we obtain a system of equations whose solutions yield estimates for the the intrinsic parameters  $k_x$ ,  $c_x$  and  $\gamma$ , and the slit parameter  $Z_1$ . To obtain a higher level of accuracy, Hartley *et al*, in [5], suggest data normalization and a non-linear technique based on the maximum likelihood criterion. Therefore to calibrate the pox-slits camera six world to image correspondences, at least, must be used.

### 3.2 General Case

Let us now address the case of the general x-slits camera. Assuming

$$\begin{aligned}
 C_1 &= L_{142}L_{234} - L_{134}L_{242} & C_2 &= L_{114}L_{234} - L_{134}L_{214} & C_3 &= L_{142}L_{213} - L_{113}L_{242} \\
 C_4 &= L_{134}L_{213} - L_{113}L_{234} & C_5 &= L_{114}L_{223} - L_{123}L_{214} & C_6 &= L_{123}L_{234} - L_{134}L_{223} \\
 C_7 &= L_{114}L_{242} - L_{142}L_{214} & C_8 &= L_{123}L_{213} - L_{113}L_{223} & C_9 &= L_{114}L_{213} - L_{113}L_{214} \\
 C_{10} &= L_{134}L_{212} - L_{112}L_{234} & C_{11} &= L_{112}L_{214} - L_{114}L_{212} & C_{12} &= L_{113}L_{212} - L_{112}L_{213} \\
 C_{13} &= L_{142}L_{223} - L_{123}L_{242} & C_{14} &= L_{123}L_{212} - L_{112}L_{223} & C_{15} &= L_{142}L_{212} - L_{112}L_{242} \\
 V_1 &= C_3 + C_5 & V_2 &= -c_x C_1 - k_x C_5 + k_x C_{10} + \gamma C_{13} & V_8 &= c_x C_8 & V_{20} &= c_y C_8 \\
 V_3 &= -c_x C_2 - \gamma C_3 + k_x C_9 + \gamma C_{10} & V_4 &= -c_x V_1 + k_x C_{12} + \gamma C_{14} & V_5 &= c_x C_4 + \gamma C_8 \\
 V_6 &= c_x C_6 + k_x C_8 & V_7 &= -c_x C_7 + k_x C_{11} + \gamma C_{15} & V_9 &= k_x C_4 + \gamma C_6 & V_{10} &= k_x C_6 \\
 V_{12} &= -k_y C_3 + k_y C_{10} - c_y C_2 & V_{16} &= k_y C_{13} - c_y C_1 & V_{17} &= k_y C_{15} - c_y C_7 & V_{13} &= k_y C_4 \\
 V_{11} &= \gamma C_4 & V_{15} &= k_y C_8 + c_y C_4 & V_{14} &= k_y C_6 & V_{18} &= k_y C_{14} - c_y V_1 & V_{19} &= c_y C_6
 \end{aligned}$$

without loss of generality, equation (2) can be rewritten as

$$P^T \begin{bmatrix} -V_{10} & -V_9 & xC_1 + V_2 & xC_6 - V_6 \\ 0 & -V_{11} & xC_2 + V_3 & xC_4 - V_5 \\ 0 & 0 & xC_7 + V_7 & xV_1 + V_4 \\ 0 & 0 & 0 & xC_8 - V_8 \end{bmatrix} P = 0$$

and

$$P^T \begin{bmatrix} 0 & -V_{14} & yC_1 + V_{16} & yC_6 - V_{19} \\ 0 & -V_{13} & yC_2 + V_{12} & yC_4 - V_{15} \\ 0 & 0 & yC_7 + V_{17} & yV_1 + V_{18} \\ 0 & 0 & 0 & yC_8 - V_{20} \end{bmatrix} P = 0$$

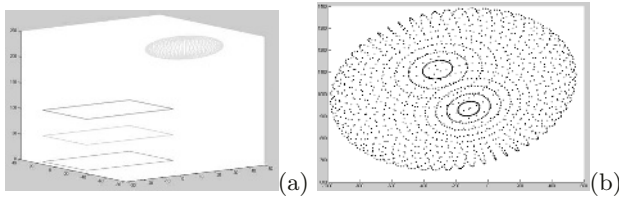
The general model of this camera is specified by 15 parameters and therefore the total number of unknowns is also 15. However, as a result of rewriting the equations so that a linear numerical method can be used, we end up with 26 unknowns. Therefore at least 13 world to image correspondences are required.

## 4 Experimental Results

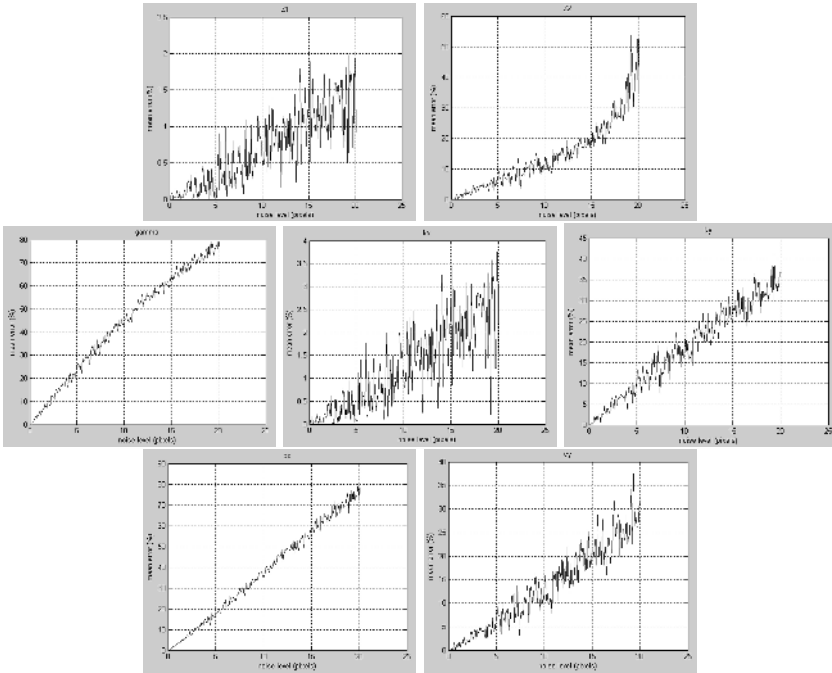
The experimental results presented in this paper use synthetically generated data. In addition we only present results for the case of the pox-slits model. Results for the general case are still being obtained.

As it can be seen in figure 2(a), a sphere is used as calibration object. This random sphere, with radius 20 and center  $(-22.33, 43.37, -226.93)$ , is made up of 1891 3D known points. In the figure we also show the image plane (bottom) and the planes that contain the slits (the two upper planes). Figure 2(b) represents the pox-slits image of the sphere points, with resolution  $1600 \times 800$ . Using equation (3), the pox-slits camera is defined with  $Z_1 = 100$ ,  $Z_2 = 50$ ,  $k_x = 47$ ,  $k_y = 63$ ,  $c_x = 320$ ,  $c_y = 240$  and  $\gamma = 25$ .

To calibrate the camera we start by normalizing the image coordinates as suggested by Hartley [5]. Gaussian white noise with 0 mean and  $\sigma^2$  variance



**Fig. 2.** (a) Visualization of the calibration object, with the image plane and the planes that contain the slits; (b) Pox-slit image of calibration object.



**Fig. 3.** Relative mean error in the estimation of the camera parameters plotted as function of the noise variance.

was added to the image coordinates of the points. The noise variance was varied between 0.1 pixels and 20 pixels. For each value of noise variance 150 runs were performed. The percent error in the estimates for each parameter was computed. The averages (for each noise variance level) of the percent errors are presented in Figure 3. As it can be seen in the figure, errors increase almost linearly with the noise level. We also computed the variance of errors in the estimates of the parameters. The values of the error variances are below the floating point precision. Therefore we can assume that this algorithm can be used to estimate this type of camera.

## 5 Conclusions

In this paper we present a robust non-iterative linear algorithm to calibrate a pox-slits camera. The algorithm requires at least with six world to image correspondences. Normalization of the coordinates of the image points is an essential step of the algorithm.

The algorithm for the general x-slit camera is also described briefly.

## References

1. D. Feldman, A. Zomet, S. Peleg, and D. Weinshall, "Video synthesis made simple with the x-slits projection", *IEEE Workshop on Motion and Video Computing*, pp. 195-200, Orlando, December 2002.
2. M. Grossberg and S. Nayar, "A General Imaging Model and a Method for Finding its Parameters", *Proceedings of IEEE International Conference on Computer Vision*, Vancouver - Canada, July 2001.
3. R. Gupta and R. Hartley, "Linear Pushbroom Cameras", *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 19, N. 9, pp. 963-975, 1997.
4. R. Hartley and T. Saxena, "The cubic rational polynomial camera model", *Image Understanding Workshop*, pp. 649-653, 1997.
5. R. Hartley and A. Zisserman, *Multiple View Geometry*, Cambridge University Press, 2000.
6. R. Kingslake, *Optics in Photography*, SPIE Optical Eng. Press, 1992.
7. B. Newhall, "The History of Photography, from 1839 to the present day", *The Museum of Modern Art*, pp. 162, 1964.
8. S. Peleg, M. Ben-Ezra, and Y. Pritch, "Omnistereo: Panoramic Stereo Imaging", *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 23, n. 3, pp. 279-290, March 2001.
9. S. Ramalingam, S. K. Lodha and P. Sturm, "A generic structure-from-motion algorithm for cross-camera scenarios", *OMNIVIS - Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, pp. 175-186, Prague, Czech Republic, May 2004.
10. S. Seitz, "The Space of All Stereo Images", *Proceedings of the 8th IEEE International Conference on Computer Vision*, vol. 1, pp. 26-33, Vancouver - Canada, July 9-12 2001.
11. J. Semple and G. Kneebone, *Algebraic Projective Geometry*, Oxford University Press, 1979.
12. P. Sturm and S. Ramalingam, "A generic concept for camera calibration", *Proceedings of 8th European Conference on Computer Vision*, Prague - Czech, 2004.
13. D. Weinshall, M. Lee, T. Brodsky, M. Trajkovic, and D. Feldman, "New View Generation with a Bi-Centric Camera", *Proceedings of 7th European Conference on Computer Vision*, pp. 614-628, Copenhagen - Denmark, May 2002.
14. J. Yu and L. McMillan, "General Linear Cameras", *Proceedings of 8th European Conference on Computer Vision*, Prague - Czech, 2004.
15. A. Zomet, D. Feldman, S. Peleg, and D. Weinshall, "Mosaicing New Views: The Crossed-Slits Projection", *IEEE Transactions Pattern Analysis and Machine Intelligence*, pp. 741-754, 2003.

# Vision-Based Interface for Integrated Home Entertainment System

Jae Sik Chang, Sang Ho Kim, and Hang Joon Kim

Dept. of Computer Engineering, Kyungpook National Univ., Daegu, South Korea  
{jschang, shkim, hjkim}@ailab.knu.ac.kr

**Abstract.** Home entertainment systems are trending to be integrated to a single system and to be more complex and difficult to control. Due to it, the methods developed for specific entertainment system are difficult to be applied to integrated systems. Accordingly, this paper presents a vision-based interface for integrated home entertainment system. The proposed interface has two types of modes: mouse control mode and instruction mode. The first mode move mouse point and click the icons using hand motion and shape and the second make instruction by hand gestures. The proposed interface is able to make predefined several gestures mapped to several similar tasks from different entertainment systems, which reduces the number of gestures and makes the interface more intuitive.

## 1 Introduction

Because of development of home network and multimedia systems, recently home entertainment systems such as home theater, games, audios and internet service systems are growing in popularity.

The interfaces for the interaction between human and the systems have been researched [1-9]. Among these interfaces, vision based interfaces have been the center of public attention due to cheap hardware and ease to use.

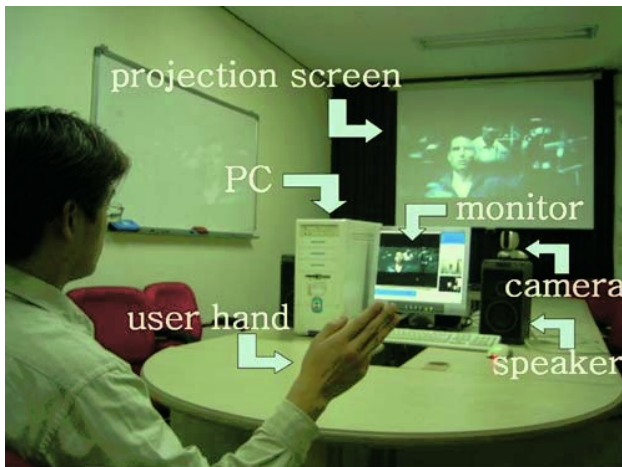
Freeman et al.[5] studied how a viewer could control a television set remotely by hand gestures. They use just a hand position to control channel and volume of a television. Lee et al.[6] implemented the PowerGesture system with which one can browse presentation program using predefined gesture commands. Shin et al.[7] described a gesture recognition system for visualization navigation. They gave an analysis of the hand motion trajectory in the registered 3-D data and classified gestures using a geometric method using Bezier curves. However, home entertainment systems are trending to be integrated to a single system and to be more complex and difficult to control. Due to it, the methods developed for specific entertainment system are difficult to be applied to integrated systems.

In this paper, we propose a vision-based interface for integrated home entertainment system. For this, the proposed interface has two types of modes: mouse control mode and instruction mode. In mouse control mode, users use their hand to move mouse point and click. The mode make user able to select an application by clicking a

icon and enjoy web surfing by control mouse point. In instruction mode, users make command to use applications such as television, games, and moving picture players. The proposed interface is able to make predefined several gestures mapped to several similar tasks from different entertainment systems, which reduces the number of gestures and makes the interface more intuitive.

## 2 System Overview

The home entertainment system using the proposed interface is shown in Fig. 1. User can control the home entertainment system with hand gestures in front of camera without any hand-held device. The distance between user and camera is about 2~3 miter. The system was implemented using a PC and a web cam without additional devices such as data gloves and frame grabber boards.



**Fig. 1.** System Environment

Our entertainment system includes three entertainment applications: television, moving picture player and web browser. To control the applications, the Interface has two types of mode as mouse control mode and instruction mode.









### 2.1 Mouse Control Mode

The mouse control mode is used for selecting one of the applications and surfing web. In the mode, open hand position control mouse point and the hand closing acts as mouse click. To select an application, user click the application icon in the user interface linked the entertainment application that user wants to use. Web browsing can be achieved by the same way. User can use the functions of web browser and hyper-links by mouse pointing and clicking.

## 2.2 Instruction Mode

The instruction mode is used to control applications such as moving picture player and television. In the mode, the functions of the applications can be used by hand gesture instructions shown in Table 1. The interface use the hand shape, open and close, to separate meaningful gestures and unintentional movements. Therefore, for control the applications, user can act the predefined gestures with open hand.

**Table 1.** Each entertainments' defined gestures

defined gestures	moving picture and DVD player	TV application
	play	
	stop	
	temporary stop	
	volume up	volume up
	volume down	volume down
	next content	channel up
	before content	channel down
	exit	exit

## 3 Vision Based Interface

To provide user-friendly remote controls, the gesture recognition system should have real-time interaction and good recognition performance across a variety of users. For this, we use hand position, shape and motions.

In the mouse mode, we use hand position and shape in an image. To estimate hand position and shape in a frame captured from camera, Skin color regions are extracted using skin color model described by 2D Gaussian model in chromatic color space [10].

In the instruction mode, we use predefined gestures as a meaningful sequence of the right hand motion [6]. The motion of the hand is defined as inter-frame position change of its region. The motion is quantized to one of the symbols which mean 8 directions. To identify the beginning and the end of the gesture, we use the shape of the hand. Hand shapes, open and close, are easily classified using size of hand region. Hand opening and closing indicate the beginning and the end of the gesture, respectively. For modeling and recognizing the gestures, we use HMMs which are robust to analyze and describe sequential data have spatiotemporal variability. Fig. 2 shows the overview of gesture recognition.

### 3.1 Hand Extraction

In this section, we describe the method for extracting hand region from a color image. The proposed method detects skin color pixels from a color image using skin color

model described by 2D Gaussian model in chromatic color space. To remove noises, morphological operations are used. Then bounding boxes of regions composed of connected components are generated. Finally, among the bounding boxes, the largest area is considered as the face, and the second and third ones are considered as hands. Fig.3 shows the results of the procedure.

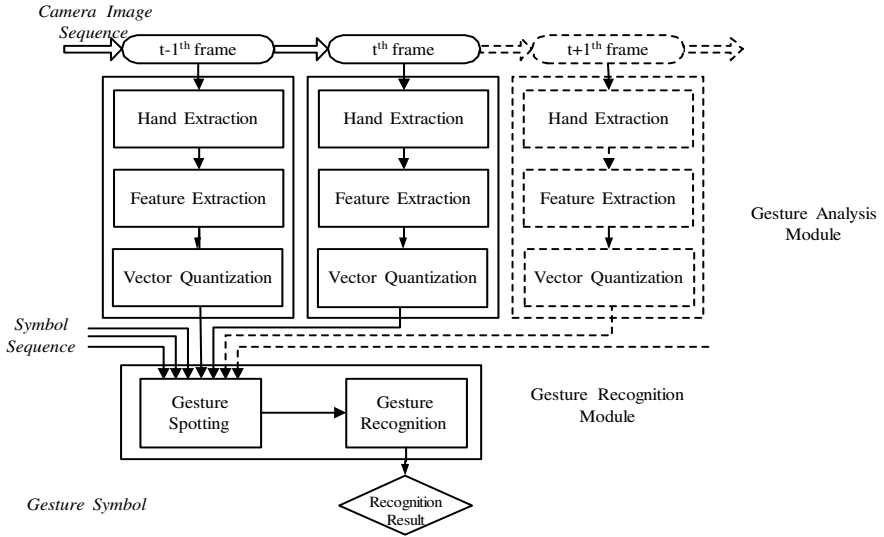


Fig. 2. Overview of of gesture recognition

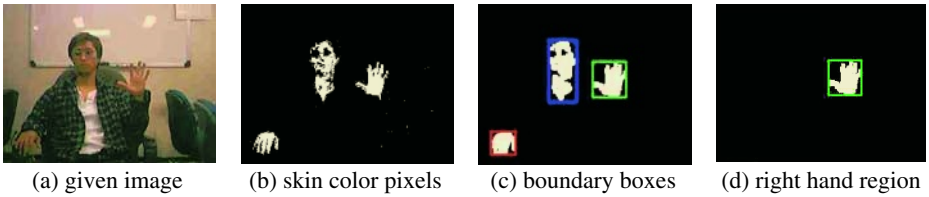


Fig. 3. Hand extraction results

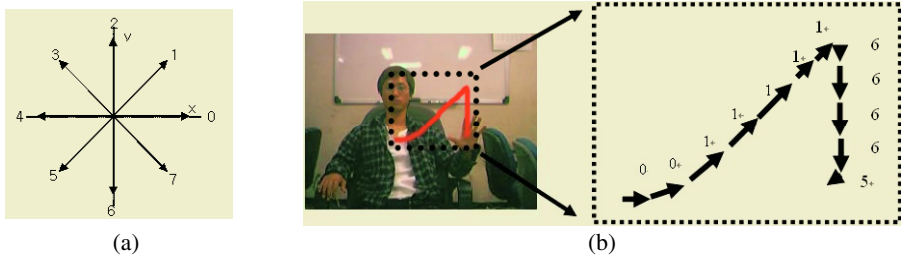


Fig. 4. “Play” gesture and codeword sequence



### 3.2 Feature Extraction and Vector Quantization

To recognize the user gestures, we used motion of right. The motion of the hand is defined as inter-frame position change of its region. We convert the feature vector to one of the 8directional codewords shown in Fig. 4(a). Accordingly, a given image is represented as a symbol and a gesture is represented as symbol sequence. Fig. 4(b) shows trajectory of “Play” gesture and extracted codewords.

### 3.3 Gesture Recognition

We defined gestures as a meaningful sequence of the open right hand motion. So we need to identify the beginning and the end of the gesture. Gestures begin when open hand appears and end when close hand appears. Open and close hand is easily classified using size of hand.

To recognize gestures, we use left-right HMMs. Given a symbol sequence, the recognizer finds the best gesture model. A gesture is recognized if the likelihood of the best gesture model is higher than the threshold value. The likelihood is estimated using forwarding algorithm [11, 12, 13]. Each gesture model consisted of five states in the left-right model and the number of state determined by experiments. Training of the HMMs followed the Baum-Welch re-estimation formulas [11, 12, 13]. Given any finite observation sequence as training data, we choose the parameters of 8 gesture models.

## 4 Experiment and Result

The interface was implemented using MS Visual C++ 6.0 and OpenCV beta3.1 to get 320×240 and 24-bit color images captured 15 frames/s without an additional frame grabber board. Fig. 5 shows the user interface. At the top, there is an image display showing the captured image from the camera. And just below there is an image display showing the hand tracked image. This allows the user to see and keep his/her hand within the camera’s field of view. At the left bottom, there are a result display and application icons. The result display is reporting the recognized gesture and gesture start and end. Application icons are composed of three type icon: movie picture player, TV, and web browser.

The skin-color model is obtained from 200 sample images. Means and covariance matrix of the skin color model are as follows:

$$m = (\bar{r}, \bar{g}) = (117.588, 79.064),$$

$$\Sigma = \begin{bmatrix} \sigma_r^2 & \rho_{X,Y} \sigma_g \sigma_r \\ \rho_{X,Y} \sigma_r \sigma_g & \sigma_g^2 \end{bmatrix} = \begin{bmatrix} 24.132 & -10.085 \\ -10.085 & 8.748 \end{bmatrix}.$$

The proposed interface was evaluated through testing 4 persons in the mouse control mode and in the instruction mode. In instruction mode, each person tried many times to perform each gesture. In the mouse control mode, each person tried many times to move mouse cursor to target and then click.

In the mouse control mode task, the gestures are succeeded in performing when person moves mouse cursor to target and then click at a time. Each 4 persons attempted it 20 times. Table 2 shows the gesture recognition performance in the mouse control mode.

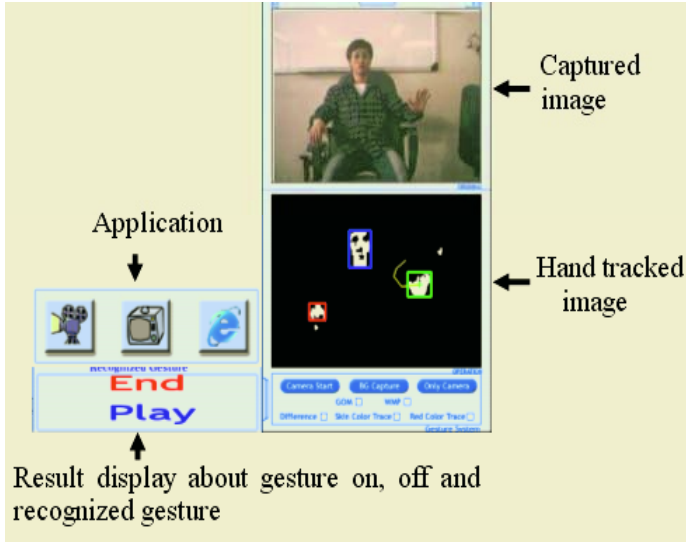


Fig. 5. User interface

Table 2. Gesture recognition results in the instruction mode

Number of attempt	success	Success ratio(%)
80	62	77.50

In the instruction mode task, we estimated gesture recognition performance by Lee et al. [6]’s test method. There are three types of errors: The insertion error occurs when the recognizer reports a nonexistent gesture, the deletion error occurs when the recognizer fails to detect a gesture, and the substitution error occurs when the recognizer falsely classifies a gesture. The detection ratio is the ratio of correctly recognized gestures over the number of input gestures as follows:

$$\text{Detection ratio} = \frac{\text{correctly recognized gestures}}{\text{input gestures}} \tag{1}$$

In calculating the detection ratio, the insertion errors are not considered. The insertion errors are likely to cause the deletion errors or the substitution errors because they often force the recognizer to remove all or part of the true gestures from observation. To take into account the effect of the insertion errors, another performance measure, called reliability, is introduced that considers the insertion errors. The reliability ratio is the ratio of correctly recognized gestures over the number of input gestures and insertion errors. Reliability is as follows:

$$\text{Reliability} = \frac{\text{correctly recognized gestures}}{\text{input gestures} + \text{insertion errors}}. \quad (2)$$

Consequently, Table 3 shows the gesture recognition performance in the instruction mode. The experiment showed 91.13 percent detection ratio and 88.33 percent reliability. In Table 3, 'I' is the insertion errors, 'D' is the deletion errors, and 'S' is the substitute errors.

**Table 3.** Gesture recognition results in the instruction mode

command	Number of gestures	correct	Error type			Detection (%)	Reliability (%)
			I	D	S		
Play	78	73	2	0	3	93.59	91.25
Stop	80	74	2	1	3	92.50	90.24
Temporary Stop	78	70	2	2	4	89.74	87.50
Volume Up	82	74	3	1	4	90.24	87.06
Volume down	77	68	4	1	4	88.31	83.95
Next Content	80	73	3	1	3	91.25	87.95
Before Content	79	72	2	2	3	91.14	88.89
Exit	77	71	2	1	3	92.21	89.87
Total	631	575	20	9	27	91.13	88.33

## 5 Conclusions

This paper presents a vision-based interface for integrated home entertainment system. The proposed interface has two types of modes: mouse control mode and instruction mode. The first mode move mouse point and click the icons using hand motion and shape and the second make instruction by hand gestures. The proposed interface was able to make predefined several gestures mapped to several similar tasks from different entertainment systems, which reduces the number of gestures and makes the interface more intuitive. Experimental results showed that the proposed interface is robust to integrated home entertainment system include several applications.

## Acknowledgement

This work is financially supported by the Ministry of Education and Human Resources Development(MOE) and the Ministry of Commerce, Industry and Energy (MOCIE) through the fostering project of the Industrial-Academic Cooperation Centered University.

## References

1. Pavlovic, V. I., Sharma, R., Huang, T. S.: Visual interpretation of hand gestures fro human-computer interaction: a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 19, No. 7 (1997) 677-695
2. Sharma, R., Zeller, M., Pavlovic, V. I., Huang, T. S., Lo, Z., Chu, S., Zhao, Y., Philips, J. C., Schulten, K.: Speech/Gesture Interface to a Visual-Computing Environment. *IEEE Computer Graphics and Applications*. (2000) 29-37
3. Quek, F.: Toward a vision-based human gesture interface. *Proceedings of International Conference on Virtual Reality Software and Technology*. (1994) 17-31
4. Rajeev Sharma, and Thomas S.Huang.: Toward Multimodal Human-Computer Interface. *Proceeding s of the IEEE*. Vol. 86, No. 5 (1998) 853-869
5. William T. Freeman, and Craig D. Weissman.: Television control by hand gestures. *IEEE International. Workshop. on Automatic Face and Gesture Recognition*. (1995) 179-183
6. Hyeon-Kyu Lee and Jin H. Kim.: An HMM-Based Threshold Model Approach for Gesture Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 21, No. 10 (1999) 961-973
7. Min C. Shin, Lenonid V. Tsap, and Dmitry B. Goldgof.: Gesture Recognition using Bezier curves for visualization navigation from registered 3-D data. *Pattern Recognition*. Vol. 37, No. 5 (2004) 1011-1024
8. Markus Kohler.: Vision Based Remote Control in Intelligent Home Environments. *3D Image Analysis and Synthesis '96*. (1996) 147-154
9. Markus Kohler.: Special Topics of Gesture Recognition Applied in Intelligent Home Environments. *Lecture Notes in Computer Science*. (1997) 285-296
10. J. Yang and A. Waibel.: A real-time face tracker. *Proceedings of the Third IEEE Workshop on Applications of Computer Vision*. (1996) 142-147
11. X.D. Huang, Y. Ariki, and M.A. Jack.: Hidden Markov Models for Speech Recognition. Edinburgh, Edingburgh Univ. Press (1990)
12. L.R. Rabiner.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEE*. Vol.77 (1989) 257-285
13. L.R. Rabiner.: An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*. Vol. 3, No. 1 (1986) 4-16..

# A Proposal for a Homeostasis Based Adaptive Vision System\*

Javier Lorenzo-Navarro, Daniel Hernández,  
Cayetano Guerra, and José Isern-González

University of Las Palmas de Gran Canaria  
Inst. Univ. de Sistemas Inteligentes y Aplic. Num. en Ingeniería  
Edif. Parque Tecnológico. Campus de Tafira, 35017 Las Palmas, Spain  
jlorenzo@iusiani.ulpgc.es

**Abstract.** In this work an approach to an adaptive vision system is presented. It is based on a homeostatic approach where the system state is represented as a set of artificial hormones which are affected by the environmental changes. To compensate these changes, the vision system is endowed with *drives* which are in charge of modifying the system parameters in order to keep the system performance as high as possible. To coordinate the drives in the system, a supervisor level based on fuzzy logic has been added. Experiments in both controlled and uncontrolled environments have been carried out to validate the proposal.

## 1 Introduction

The performance of most computer vision applications relies heavily on the “quality” of the images supplied by the acquisition subsystem, normally a video camera. But this “quality” is influenced by factors as hardware, camera and acquisition board, lighting conditions, size and position of the object of interest and many others. The variations of some of these factors can be limited for some tasks as machine vision or indoor applications. However there exists more challenging computer vision applications where some of the previous factors can not be controlled as mobile robot applications and indeed human computer interaction in indoor scenarios. So it is necessary to endow these systems with mechanisms which allow them to survive in environments where the conditions can vary in a wide range of values.

In nature, living beings can survive in a world where the environmental conditions are continuously changing and they can perform their tasks with success. Homeostasis is one of the mechanisms that the living beings own to adapt their behavior to the environmental changes. Homeostatis is defined in the Merriam Webster on line dictionary as “a relatively stable state of equilibrium or a tendency toward such a state between the different but interdependent elements or

---

\* This work has been partially supported by the Spanish Ministry of Education and Science and FEDER funds under research project TIN2004-07087, the Canary Islands Regional Government under projects PI2003/165 and PI2003/160 and the University of Las Palmas under projects UNI2003/10, UNI2004/10 and UNI2004/25.

groups of elements of an organism, population, or group". The state of equilibrium is normally related to the survival of the animal in an environment making sure that it gets enough to eat or it does not overheat or freeze.

The idea of homeostasis has been introduced by some authors in the construction of systems that have to develop their activity in complex environments. Arkin and Balch [1] propose a homeostatic regulation system which modifies the performance of the overall motor response according to the level of internal parameters such as battery or temperature. Another work which includes a homeostatic regulation mechanism is the proposal of Hsiang [2] who introduces it to regulate the dynamic behavior of the robot during task execution.

The works reviewed above are mainly related to robotics since robots possess effectors to act on the environment, but we have none tackling the introduction of homeostasis in a vision system. However, since the introduction of the Active Vision paradigm [3], vision systems include perception strategies which are controlled by the interaction with the environment when a specific goal is pursued. Thus, we can consider the introduction of a homeostatic regulation in such vision systems because they share with the previously described systems the fact that a goal has to be achieved (survive) in a changing environment and they have to adjust their behavior in order to get always the best possible performance.

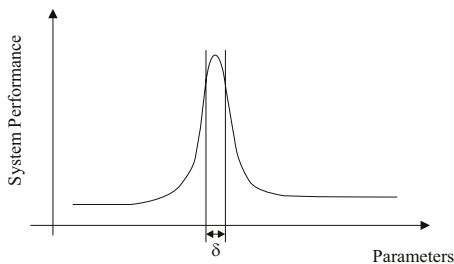
Some important considerations must be taken into account when putting homeostatic regulation into practice. Initially, a homeostasis regulation mechanism can be configured as a set of independent drives operating at a predefined frequency. However, in practice the execution of some drives can affect others requiring a certain level of coordination to avoid undesired effects. Additionally, active-vision and mobile robotic applications are usually conceived as tactical multipurpose systems. This requires an implementation based on multiple periodic tasks executing concurrently on systems with limited resources [4]. If not correctly managed, this contention could lead to poor performance, threaten system security or even block it, when in competition for CPU time, for example [5]. Thus our base homeostatic regulation level must incorporate a higher supervisor level.

Some alternatives that have been proposed for computational adaptation include any-time processing scheme [6], imprecise computation [7] or variable frequency [8]. In our context, adaptation should deal with several aspects such as drives coordination, inter-level coordination, priority-based degradation, and resource management (CPU processing time, memory, energy). For this purpose we have selected a fuzzy inspired adaptation control that complements the homeostatic regulation.

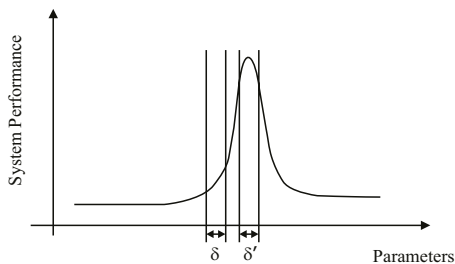
This paper explores the introduction of a homeostatic adaptive mechanism in a computer vision system based on artificial hormones which are regulated by means of drives, this first level of adaptation is described in Section 2. Section 3 considers the introduction of a higher level of adaptation based on fuzzy rules to take into account the possible interdependences among drives. Finally, in Section 4 the experiments carried out with an implementation of the architecture proposed here are presented.

## 2 The Homeostatic Regulation Mechanism

In computer vision applications where the environment  $E$  is completely controlled, i.e. industrial applications, the camera parameters that define the quality of the image are initially tuned to get the best performance. This is illustrated in Figure 1 where the set of camera parameters  $\delta$  is the one which maximizes the performance of the system under the environmental conditions  $E$ . If the environment changes to  $E'$ , for example due to different lighting conditions, the performance of the system will be maximum for another set of camera parameters  $\delta'$  as it is shown in Figure 2. So if the system does not have an internal mechanism to detect the new environment  $E'$ , its performance diminishes because it will continue using the initial parameter setting  $\delta$ , and we must rely on an external agent to readjust the parameter setting to  $\delta'$ .



**Fig. 1.** Setting of camera parameters for an environment  $E$

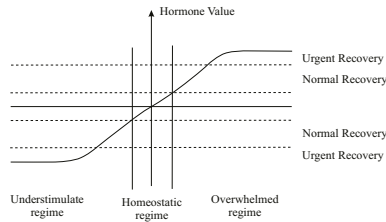


**Fig. 2.** Setting of camera parameters for an environment  $E'$

In order to adopt in our proposal the affective computing framework [9] which establishes that systems must be “bodily” because human emotions involve both the body and the mind, we simulate the physiological changes that affect the homeostasis mechanism. Cañamero [10] proposes synthetic hormones to imitate physiological changes in the body of a robot which evolves in a two-dimensional world and the motivations of the robot respond to the levels of the synthetic hormones. We adopt this approach in our system and implement a set of synthetic hormones that reflect the internal state of the system “body”.

The homeostatic mechanism is governed by the value of the hormones which are computed from the controlled parameter by means of a sigmoid mapping (Fig. 3). In this way, adaptive strategies can be implemented more easily in the drives defining normal and urgent recovery zones which are independent of the range of values of the controlled parameter [11]. In our system the hormones are associated to the image luminance (h\_luminance), contrast (h\_contrast), white balance (h\_whitebalance) and size of the object (h\_size).

The luminance of the image is controlled by dividing the image into five regions similarly to the method proposed by Lee et al. [12]. This image partition allows different luminance control strategies by giving different weights to the average luminance in each region according to the nature of the task. To compute the contrast of the image, a measure [13], which exhibits a maximum when the



**Fig. 3.** Hormone value mapping from the variable of interest

image is at the best focus proposed, was used in an auto-focus algorithm that obtain the best focus position avoiding a hill-climbing search [11]. For white balance, we adopt the *Grey World* [13] assumption which tries to make the average amount of green, blue and red in the image equal, by adjusting the red and blue gain parameters. Finally to control the size of the object in the image we act on the zoom of the camera.

As previously stated, an important element in a homeostatic mechanism is its adaptive aspect. When the internal state of the body is too far away from the desired regime, the homeostatic mechanism must recover it as soon as possible, giving less priority to other tasks if it is necessary. To accomplish this, we have included a higher level in the proposed architecture, that will be described in the following section.

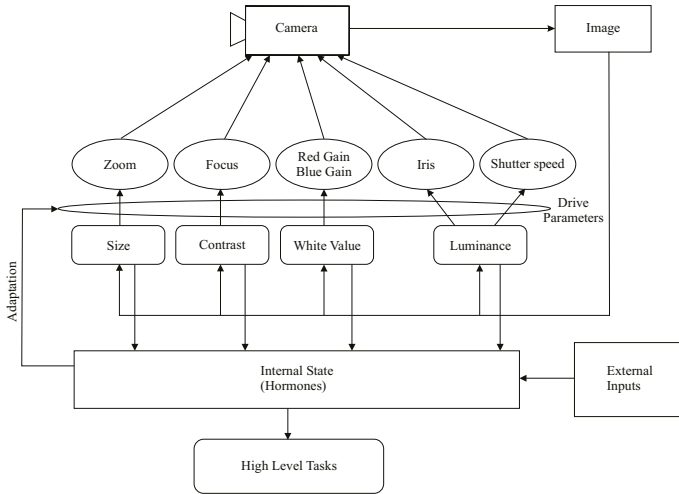
### 3 Rule Based Coordination Level

At low level, homeostatic drives should be coordinated to take into account interdependencies, as several homeostatic drives are executing simultaneously it can produce side effects on each other that make the settling times larger than if execution sequence is supervised. In other cases, simply it makes not sense executing some drives when others are far out from their desired regime values (e.g. focusing on a very dark image). Additionally, some high processing level tasks depend on the stabilization of the homeostatic level to produce valid results, so their execution should be conditioned to this situation.

On the other hand, if the vision system is on a mobile robot, regulation mechanism must deal with a multiple-task shared-resource system. The global system operation normally requires the execution of multiple homeostatic drives as well as high-level application tasks concurrently. In case of resource shortage, low priority tasks have to be slowed-down or postponed, releasing resources for higher priority tasks. Some examples include execution on a saturated CPU or low-battery conditions.

Thus, the basic homeostatic mechanism described previously has been complemented with a higher level in order to improve performance. Figure 4 shows the architecture proposed combining both regulation levels, homeostatic and rule based level. It can be noticed that homeostatic drives run independently according to the values of the hormones associated to the measures obtained





**Fig. 4.** Elements of the homeostatic adaptive regulation mechanism

from the image. A fuzzy-based approach has been selected for supervision, as we consider it specially suitable for this vision based context, is implemented and it takes into account the global state of the system to modify the operation of the homeostatic drives.

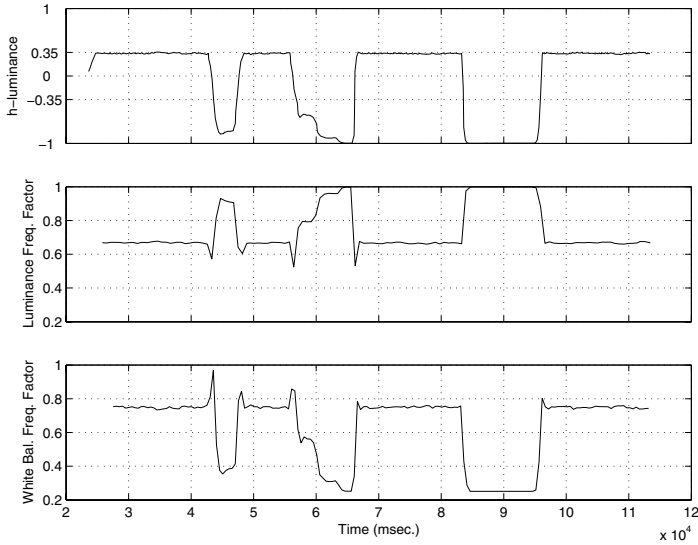
The whole regulation architecture proposed in this work has been based on the configuration of each task in the system as a periodic process, with a desired frequency of operation to be met whenever possible; this includes both homeostatic drives and high level application tasks. So the upper adaptive level modifies the operation periods of the tasks by means of frequency commands, allowing a modification on resource demands such us CPU processing time. Although other actions can be generated as quality commands, they have not been considered in this implementation.

The rules implemented in this work take the form of fuzzy implications with conditions on state system (hormones) as antecedents, and actions on system tasks as consequents. A rule is characterized by a priority value and a method to combine the certainty of each premise to give the certainty of the rule (minimum, mean, product). Additionally, the action part is defined by the type of control action and its target. Some examples of these rules are the following ones:

R1: High Priority IF `h_luminance` is not zero THEN Decrease white balance drive operation frequency

R2: Normal Priority IF `h_whitebalance` is zero THEN Decrease white balance drive operation frequency

R3: Normal Priority IF `h_luminance` is zero THEN Decrease luminance drive operation frequency



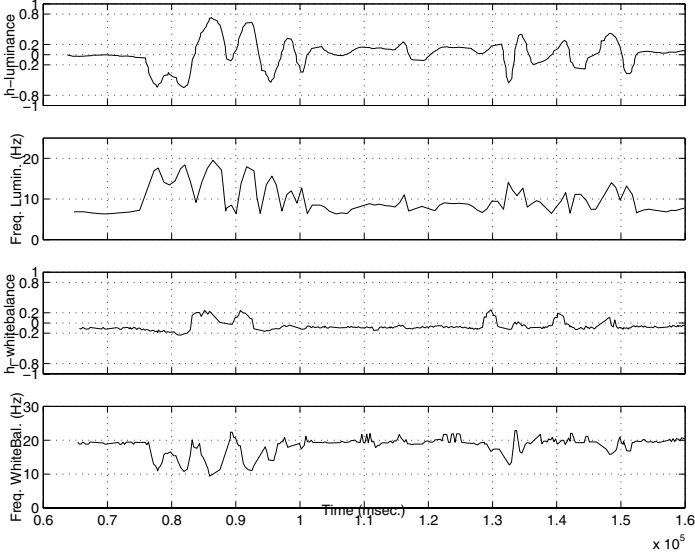
**Fig. 5.** Evolution of the operation frequency in drives depending of  $h\_luminance$  hormone value for the first experiment

In previous rules **zero** is a fuzzy linguistic label whose membership function can be obtained from the hormone value which is bounded in the interval  $[-1,+1]$  (Fig. 3). The highest priority rule, **R1**, is responsible of giving the most CPU resources to the recovery of the luminance hormone when it is out of the homeostatic regime. The other two rules, **R2** and **R3**, relax the drives associated to the luminance and white balance hormones when they have their desired values, reducing the load of the CPU that can be assigned to other tasks in the system.

## 4 Experiments

Some tests have been performed to evaluate the proposal presented in this paper. A first bunch of experiments was realized in a controlled environment with fixed lighting conditions to validate the obtained results against the expected behavior of the system. In these experiments we used a static firewire color camera and changed the luminance of the object of interest. The application consists on two hormone drives controlling simultaneously luminance and white balance hormones of the image and a set of fuzzy rules to change the operation frequency of each drives according to the luminance hormone value.

In Figure 5, the value of the luminance hormone is shown together with frequency degradation factors for luminance and white balance drives (1 means no degradation). As it is shown, when hormone value separates from homeostatic regime (centered in 0), luminance drive runs faster to recover image quality as soon as possible, while white balance drive slows down its operation because



**Fig. 6.** Hormone values and operation frequencies of the associated drives for the second experiment

it makes no sense to recover the white balance until luminance is close to its homeostatic regime. When luminance is close to the desired value, the associated drive can relax and the white balance drive recover its normal operation frequency.

A second experiment was carried out making use of a mobile robotic platform performing a line following task in a uncontrolled environment with changing lighting conditions, with a specially dark area near the beginning of the path due to the existence of a kind of tunnel that the robot must traverse, so that, without homeostasis the robot task fails.

In Figure 6, the luminance hormone values are represented together with execution frequency values for luminance and white balance drives. As it is shown, when luminance hormone goes far from homeostatic regime, luminance drive runs faster to recover image quality as soon as possible, while white balance becomes slower. This effect becomes more noticeable when traversing the dark zone, between seconds 75 and 100. In homeostatic regime, white balance drive is allowed to execute at a higher frequency while luminance drive gets relaxed, for example when the robot is on the first straight segment and the first curve (seconds 100 to 125). The robot velocity is also governed by the image quality to avoid losing the desired path.

## 5 Conclusions

The introduction of the homeostatic regulation mechanism improves the performance of an active vision system, as the mean quality of the sensor data increases

in dynamic environments. The combination of this low-level adaptation mechanism with a high-level fuzzy adaptive control has exhibited a better outcome under variable run-time conditions. The result, as illustrated in the experiments, is a highly-configurable framework that improves the system performance and extends its range of operation.

## References

1. Arkin, R.C., Balch, T.: AuRA: Principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence* **7** (1997) 175–188
2. Hsiang, K., Kheng, W., Ang, M.: Integrated planning and control of mobile robot with self-organizing neural network. In: 18th IEEE Int. Conference on Robotics and Automation, Washington DC (2002) 3870–3875
3. Aloimonos, J.Y.: Introduction: Active vision revisited. In Aloimonos, J.Y., ed.: *Active Perception*. Lawrence Erlbaum Assoc. Pub., New Jersey (1993)
4. D'Ambrosio, B.: Resource bounded-agents in an uncertain world. In: *Proc. of the Workshop on Real-Time Artificial Intelligence Problems*, Detroit, MI, USA (1989)
5. Stewart, D.B., Khosla, P.K.: Mechanisms for detecting and handling timing errors. *Communications of the ACM* **40** (1997) 87–93
6. Zilberstein, S.: Using anytime algorithms in intelligent systems. *AI Magazine* **17** (1996) 73–83
7. Liu, J., Lin, K., Bettati, R., Hull, D., Yu, A. In: *Use of Imprecise Computation to Enhance Dependability of Real-Time Systems*. Kluwer Academic Publishers (1994) 157–182
8. Garvey, A.J., Lesser, V.: Design-to-time real-time scheduling. *IEEE Trans. on Systems, Man and Cybernetics* **23** (1993) 1491–1502
9. Picard, R.W.: *Affective Computing*. The MIT Press, Cambridge, Massachusetts (1997)
10. Cañamero, D.: Modeling motivations and emotions as a basis for intelligent behavior. In Lewis, J., ed.: *Proceedings of the First Int. Symposium on Autonomous Agents*, New York, ACM Press (1997) 148–155
11. Lorenzo, J., Castrillón, M., Hernández, M., Déniz, O.: Introduction of homeostatic regulation in face detection. In Fred, A., ed.: *Proceedings of the 4th International Workshop on Pattern Recognition in Information Systems, PRIS 2004*, Porto (Portugal) (2004) 5–14
12. Lee, J.S., Jung, Y.Y., Kim, B.S., Sung-Jea, K.: An advanced video camera system with robust AF,AE and AWB control. *IEEE Transactions on Consumer Electronics* **47** (2001) 694–699
13. Nanda, H., Cutler, R.: Practical calibrations for a real-time digital omnidirectional camera. In: *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR 2001)*. (2001)

# Relaxed Grey-World: Computational Colour Constancy by Surface Matching

Francesc Tous, María Vanrell, and Ramón Baldrich

Computer Vision Center, Dept. Informàtica  
Universitat Autònoma de Barcelona  
08193 Bellaterra (Barcelona), Spain  
{ftous,maria,ramon}@cvc.uab.es

**Abstract.** In this paper we present a new approach to computational colour constancy problem based on the process of surface matching. Classical colour constancy methods do not usually rely on this important source of information and they often use only partial information in the images. Our proposal is to introduce the use of a set of canonical surfaces and its matching versus the content of the image using a ‘relaxed’ grey-world assumption to perform colour constancy. Therefore, our approach takes into account information not considered in previous methods, which normally rely on statistical information in the image like highest luminance or image gamuts. Nevertheless the selection of the canonical surfaces is not a trivial process and should be studied deeply.

## 1 Introduction

The human visual system has the capability to perceive the same colour for a given surface regardless the colour of the illuminating light. This is a fundamental property to colour vision and pursues the perception of a stable coloured world, even though the stimulus reaching the retina differs for the same surface under different conditions of illumination. The perceived colour of a white patch under a blue sky compared to the same patch in a room with a light bulb is perceived as the same colour. Actually in the first situation the reflected light reaching the eye has a bluish spectrum compared to the reddish reflected light of the second. This ability is known as colour constancy, the constant appearance of surface colours despite changes in the colour of the illumination. The mechanisms of human colour constancy have not yet been completely understood, and there are different approaches trying to explain them [1–4].

## 2 Background

RGB images are formed by the light reflected from different surfaces reaching three sensors that integrate the incident light at different wavelengths. The color of a surface depends on the surface reflectance and the colour of the incident light. The aim of computational colour constancy is to find an illuminant invariant description of a scene from an image taken under unknown lighting conditions.

This process is often performed in two steps: (1) estimate the illuminant parameters and (2) use those parameters to build illuminant independent description of the scene. For these methods a canonical illuminant must be defined, i.e. an illuminant for which the camera is balanced and the colours appear in a trustworthy form. Under this illuminant, the RGB values of an image of a scene can be taken as descriptors of the surfaces. There is a wide literature on computational colour constancy methods [5–10]. None of them performs perfectly on all kind of images under weak assumptions.

Many of these methods directly estimate the illumination change from the unknown illuminant to the canonical illuminant. Considering the von Kries adaptation model [11], the transform of an illuminant change can be modelled by a linear diagonal model, as proven in [12]. For example, the RGB response of a camera to a white patch under an unknown illuminant is  $(R_w^U, G_w^U, B_w^U)$  and the response under the canonical illuminant is  $(R_w^C, G_w^C, B_w^C)$ , the illuminant change from the unknown to the canonical illuminant can be obtained by scaling the three channels by  $R_w^C/R_w^U, G_w^C/G_w^U, B_w^C/B_w^U$  respectively. Thus, the colour of the illuminant of an RGB image can be modified by a diagonal change (1),

$$(R^C, G^C, B^C) = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix} \begin{pmatrix} R^U \\ G^U \\ B^U \end{pmatrix} \quad (1)$$

where  $\alpha = R_w^C/R_w^U, \beta = G_w^C/G_w^U, \gamma = B_w^C/B_w^U$ . In a typical colour constancy problem, we have acquired the image under an unknown illuminant,  $(R^U, G^U, B^U)$ , and try to obtain the surface descriptors,  $(R^C, G^C, B^C)$ . The triplet  $(\alpha, \beta, \gamma)$  is called a map, and knowing the actual map implies a guessing of the unknown illuminant.

The different methods proposed in the literature can be sorted in different classes regarding the assumptions they are based on. The first family of algorithms are established upon the Retinex theory of human vision [13], which goes beyond simple illuminant estimation. The theory assumes that slight spatial changes in the response are due to changes in illumination or noise, and large changes correspond to surface changes. The idea is to run random paths from every surface and compute the ratio of the responses in each channel. The descriptor of a pixel is given by the average of the ratios from different paths beginning at the same pixel.

Another group are the Grey World methods. They are based on the assumption that the scene is colorimetrically unbiased (no particular colour predominates). In other words, supposes that a complex scene contains a wide range of reflectances, whose mean is a grey reflectance (for instance, a uniform reflectance with half of the maximum energy). Therefore, to correct the illumination of an image the map that takes the average of the image to the average of the canonical gamut is used as an estimation of the illuminant change.

One of the most important groups to date are the Gamut Mapping methods. All of them are based on the idea of canonical gamut firstly introduced by Forsyth in [5]. If we consider all the possible reflectances under a canonical

illuminant we obtain a convex set of RGB values, which are the whole set of values that can be perceived under the canonical illuminant for a given camera. This introduces a device/illuminant restriction, and it can be used to build a set of illuminant changes that are feasible, i.e. which map the image gamut within the canonical gamut. To build the feasible set of illuminant changes, the image gamut is computed first. All the maps from a single colour in the image gamut to each colour in the canonical gamut form a convex set. The intersection of the convex sets obtained for each vertex in the image gamut results in a convex set of feasible maps. This feasible set, which is given in the map space,  $\alpha\beta\gamma$ -space, normally contains a wide range of assorted maps unless the gamut of the image is large enough to reduce the possible bindings of the image gamut inside the canonical gamut. A selection step is needed to choose the optimal map inside the feasible set, i.e. the best approximation to the unknown illuminant. Different heuristics have been used to obtain a single answer. The most successful heuristic [14] is the selection of the map that maximises the volume of the mapped image gamut, i.e. the map that makes the image gamut as colourful as possible within the bounds of the canonical gamut, also known as CRULE. Other heuristics like the average map of the feasible set have also been studied. Several methods have derived from Forsyth first approach, [9, 15].

Another kind of methods are those based on Colour by Correlation which propose to study the chromaticities of an image to decide among a set of proposed illuminants the one that is more compatible with the chromaticities found [16]. A correlation matrix is pre-computed and describes for each of the selected illuminants the occurrence of image chromaticities. Each row in the matrix corresponds to a different training illuminant and matrix columns to possible chromaticity ranges.

An interesting study comparing the performance of these different methods described can be found in [14]. There are more contributions which are important in colour constancy but they do not adapt to the context we work in, as they deal with the recovery of surface spectral reflectances using reduced sets of linear bases [6].

### 3 Surface Matching

The method we propose in this paper tries to introduce the surface matching phenomenon, previously studied as one of the cues of how the human visual system performs colour constancy [4, 17], to reduce the number of possible map solutions. Nevertheless the idea has not yet been explored when performing computational colour constancy. In the process of guessing the illuminant of an image, it is likely to match the colours that we find in the image with colours that we have previously learned, which are a set of colours we already know for its significance. It can be easily assumed that when looking at an image a part of the colour constancy process is the matching of the colours that we see in the image with colours that we ‘expect’ to find in the image. This refers to a previously learned knowledge of common colours as seen under an ideal, canonical, illuminant.

Considering this idea, we can pair the colours that are present in our image with ‘reference’ colours. The values of these colours as they would be seen under the canonical illuminant can be computed and they can be named as canonical colours or ‘canonical surfaces’. Therefore, we can match every surface in our image with a ‘canonical surface’. This is the surface matching process, also known as ‘asymmetric colour matching’ and depicted in [4]. To perform the ‘surface matching’ process, we need the set of surfaces to match with. In our surface matching approach, we propose to use a reduced set of ‘canonical surfaces’, carefully selected to represent the most important and frequent colours. The selection of these canonical surfaces is a hard goal that should be addressed.

## 4 Relaxed Grey-World

Surface matching implies to match every image surface with every canonical surface, that is to generate all the possible combinations of matchings. Even using a reduced and significant set of image surfaces and a small set of canonical surfaces the set of pairs of matches that can be derived is too large and introduces lots of non-consistent pairs of matchings (if a reddish image surface is matched with a bluish canonical surface, it is not coherent to match another bluish image surface with a reddish canonical surface). This leads us to introduce an assumption to constrict the set of matchings, in order to build a consistent set losing minimum performance.

The Grey-World assumption, as depicted before, supposes the average of an image is grey. Even though this is a strong assumption it can help us to find the consistent constriction that maintains the colour structure of the image gamut. In order to relax this assumption we propose another one:

**Relaxed Grey-World Assumption.** *The image gamut under the canonical illuminant contains grey or its average is close to grey.*

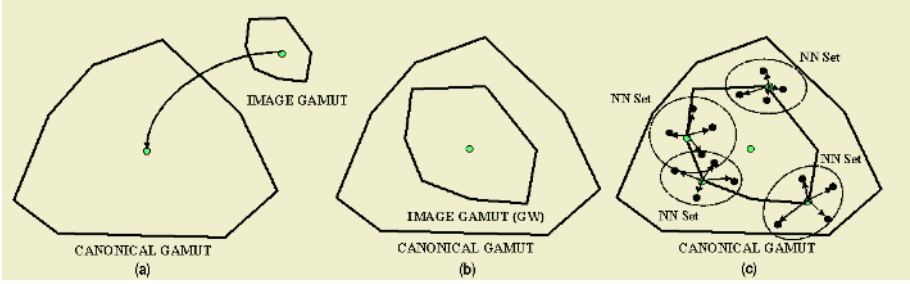
Considering this assumption the set of canonical surfaces that can be paired with each image surface can be reduced to the canonical surfaces which are close to the image surfaces when the grey-world map is applied to the image, figure 1. That is, the grey-world assumption is relaxed in order to find the solutions near the grey-world, enabling some sort of flexibility near this solution.

The relaxed grey world assumption combined with surface matching lead us to the new approach we propose in this paper. The method matches the image surfaces with canonical surfaces that we have previously selected, but only with the surfaces that are consistent with the relaxed grey-world assumption, i.e. the canonical colours near a neighbourhood in the grey world transform.

First of all we need to select a representative set of surfaces and compute their RGB values for the canonical illuminant, which is selected to be well balanced with our sensor. Hence we have a set of  $k$  canonical surfaces, denoted as  $S^C = \{S_1^C, S_2^C, \dots, S_k^C\}$ .

Thus, for a given image,  $I$ , acquired under an unknown illuminant  $U$ , the matching algorithm is carried out with the following steps:





**Fig. 1.** The relaxed grey-world assumption leads us to find a set of nearest-neighbour canonical surfaces for each image surface. The image is mapped to the center of the canonical gamut (a),(b) and there the nearest-neighbour canonical surfaces for each image surface are selected (c).

1. Getting RGB values of surfaces from the image  $I$ , denoted as  $S^U(I) = \{S_1^U, S_2^U, \dots, S_n^U\}$ , where  $n$  is the number of surfaces.
2. Applying the grey world transform to  $S^U(I)$ , which places the center of the image gamut in the center of the canonical gamut (fig. 1 a,b). It is denoted as  $S^{GW}(I)$ .
3. For each surface,  $i = 1 \dots n$ , of  $S^{GW}(I)$  we select the  $m$  nearest neighbours surfaces from the canonical surfaces (fig.1 c),  $S^C$ , we denote this set as  $S_i^{NN}$ .
4. Computing the set of all possible correspondences between each  $S_i^U$  with all the surfaces in  $S_i^{NN}$ , we name this set  $RCorr = \{S_1^U = S_{1,p_1}^{NN}, S_2^U = S_{2,p_2}^{NN}, \dots, S_n^U = S_{n,p_n}^{NN}; \forall p_i = 1, \dots, m\}$ , where  $\#RCorr = m^n$ .
5. For each element of  $RCorr$ , the corresponding  $\alpha\beta\gamma$  map is computed, and we obtain a set of maps,  $MAP_{\alpha\beta\gamma}^{RCorr}$ .
6. All the maps in  $MAP_{\alpha\beta\gamma}^{RCorr}$  out of the feasible set are removed, as we do not want to deal with impossible maps.

Once we have generated the set of maps,  $MAP_{\alpha\beta\gamma}^{RCorr}$ , we propose to use one of the existing heuristics to select one map within this set. In the following section we show the results using the heuristics of maximum gamut volume and average of the set. A simplification of the process can be seen in figure 2.

## 5 Experiments and Results

To evaluate our method in this first approach we have looked at its performance using only synthetic data. This is a first way to evaluate methods because performance is not affected by image noise and we are able to evaluate performance over hundreds of synthetic images and thus obtain a reliable performance statistic. Otherwise, with real data these problems arise, and also the available datasets are not large enough to extensively test the method.

To build the RGB of the canonical surfaces, we have chosen a synthetic planckian illuminant with CCT=6500K (fig. 3 (a)). A gaussian narrow-band sensor has been built, with centers in 450, 540 and 610 nm (fig. 3 (b)). Hence, the

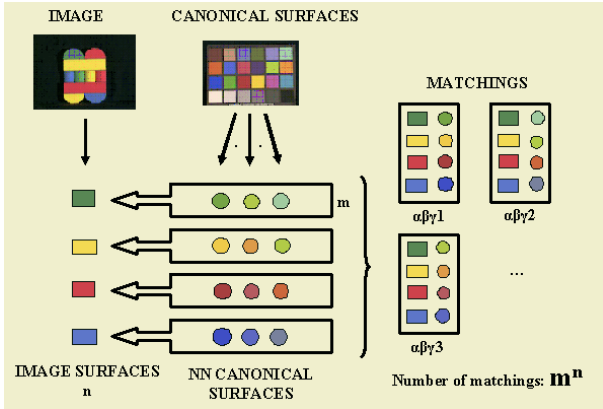


Fig. 2. An illustration of how the relaxed grey world algorithm proceeds.

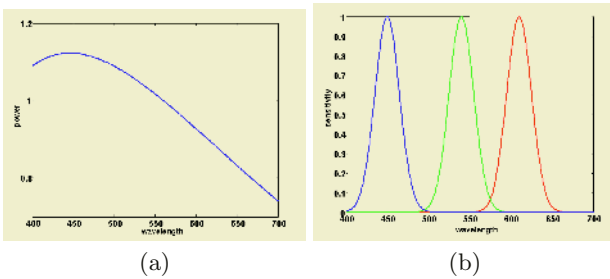


Fig. 3. The synthetic illuminant (a) and sensor (b) used in the experiments.

1995 reflectances of the Munsell chips have been used to synthesise the RGB values of our canonical set of surfaces.

Once we have selected the canonical surfaces we generate synthetic images to test the algorithm. 400 images consisting of 10 reflectances per image (from Munsell chips randomly selected) under a random illuminant, chosen from a frequently used selection of 11 different illuminants [14]. To test the method, we have selected 6 surfaces from each image and found their 5 nearest neighbours surfaces from the canonical surfaces, that is  $n = 6$  and  $m = 5$ .

We have used as recovery error the angular error between the RGB of the estimated illuminant,  $\widehat{RGB}_w^C$ , and the RGB of the canonical illuminant used,  $RGB_w^C$  (as it is done in [14]). These RGB values of the illuminants are normally unknown in real images, but they can be computed easily working with synthetic data.

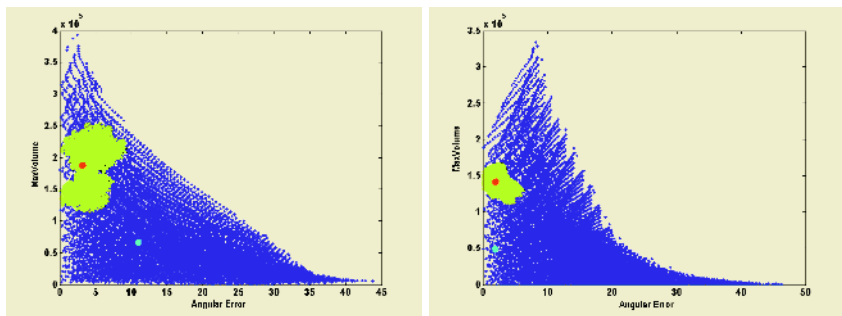
$$\text{recovery error} = \text{angle}(\widehat{RGB}_w^C, RGB_w^C)$$

In table 1 we can see the performance of the proposed method versus one of the most significant colour constancy algorithms that normally achieves best results [14], CRULE (introduced by Forsyth in [5]). The performance varies

**Table 1.** Comparison of the performance of the two methods. The value shown is the root mean square of the angular errors computed for the 400 synthetic images.

Heuristic	CRULE	Relaxed Grey-World
Maximum Volume map	7.09°	7.55°
Average map	9.35°	6.62°

depending on the heuristic used to select the optimal map within the computed maps. As it can be seen, the best performance is obtained taking the average map of the proposed Relaxed Grey World. This improvement reinforces the use of the relaxed grey-world assumption. Also, in figure 4 the different sets of maps generated with the two algorithms can be compared. With our method, we avoid to generate a large set of maps that includes the worse maps. We look for a reduced set of maps which includes the best solutions. In this sense we have computed the average value of the best angular error for each of the 400 images and it has resulted to be 1.9°, which means that an optimal map is included in our set of maps in the most of the cases. This result combined with the performance of our method using the average as heuristic justifies the use of the reduced set of maps.

**Fig. 4.** Comparison of the sets of maps generated with CRULE (dark dots) versus the set of maps generated with our method (bright dots) for 2 different images. In the x-axis is represented the angular error and in the y-axis the maximum volume heuristic.

## 6 Discussion

As it has been proven, the introduction of the surface matching approach to solve computational colour constancy opens a new line of research in this problem that can help in reducing the error of current methods, that ignore image information that can be introduced by surface matching. The method proposed performs good in the synthetic world and this encourages us to go on with its improvement. The selection of canonical surfaces is an important step to pay more attention and to be focus of a deep study. Indeed, the number of canonical surfaces used in our experiments may seem too large to depict representative colours, but it

has been used as a first approach to the surface matching method, to test how good it could perform. Further work needs to be done in the selection of the set of canonical surfaces, as they should represent more trustworthily our knowledge of colours. When done, this part of the process of colour constancy in the human visual system will be enabled to take part in computational approaches.

## Acknowledgments

This work has been partially supported by project TIN 2004-02970 (Ministerio de Educación y Ciencia).

## References

1. Land, E.H., McCann, J.J.: Lightness and retinex theory. *Journal of the Optical Society of America* **61** (1971) 1–11
2. Spitzer, H., Semo, S.: Color constancy: a biological model and its application for still and video images. *Pattern Recognition* **35** (2002) 1645–1659
3. Wandell, B.A.: *Foundations of Vision*. Sinauer Associates, Inc. (1995)
4. Foster, D.H.: Does colour constancy exist? *TRENDS in Cognitive Sciences* **7** (2003) 439–443
5. Forsyth, D.A.: A novel algorithm for color constancy. *Int. J. Comput. Vision* **5** (1990) 5–36
6. Maloney, L.T., Wandell, B.A.: Color constancy: a method for recovering surface spectral reflectance. *Journal of the Optical Society of America* **3** (1986) 29–33
7. Sapiro, G.: Color and illuminant voting. *Transactions on Pattern Analysis and Machine Intelligence* **21** (1999) 1210–1215
8. Funt, B., Ciurea, F., McCann, J.: Retinex in matlab. In: *Proceedings of the IST/SID Eighth Color Imaging Conference: Color Science, Systems and Applications*. (2000) 112–121
9. Finlayson, G.: Color in perspective. *Transactions on Pattern Analysis and Machine Intelligence* **18** (1996) 1034–1038
10. Funt, B., Barnard, K., Martin, L.: Is colour constancy good enough? In: *5th European Conference on Computer Vision*. (1998) 445–459
11. von Kries, J.: Beitrag zur physiologie der gesichtempfindung. *Arch. Anat. Physiol.* **2** (1878) 505–524
12. Finlayson, G., Drew, M., Funt, B.: Diagonal transforms suffice for color constancy. In: *ICCV93*. (1993) 164–171
13. Land, E.H.: The retinex theory of color vision. *Scientific American* (1977) 108–128
14. Barnard, K., Cardei, V., Funt, B.: A comparison of computational colour constancy algorithms. part one. methodology and experiments with synthesized data. *IEEE Transactions on Image Processing* **11** (2002) 972–984
15. Finlayson, G.D.: Color constancy in diagonal chromaticity space. In: *Proceedings of the Fifth International Conference on Computer Vision*. (1995) 218–223
16. Finlayson, G.D., Hordley, S., Hubel, P.M.: Colour by correlation: A simple, unifying framework for colour constancy. *Transactions on Pattern Analysis and Machine Intelligence* **23** (2001) 1209–1221
17. Troost, J.M., Weert, C.M.D.: Naming versus matching in color constancy. *Perception and Psychophysics* **50** (1991) 591–602

# A Real-Time Driver Visual Attention Monitoring System

Jorge P. Batista

ISR-Institute of Systems and Robotics, DEEC/FCT  
University of Coimbra, Coimbra, Portugal  
batista@isr.uc.pt

**Abstract.** This paper describes a framework for analyzing video sequences of a driver and determining his level of attention. The proposed system deals with the computation of eyelid movement parameters and head (face) orientation estimation. The system relies on pupil detection to robustly track the driver's head pose and monitoring its level of fatigue. Visual information is acquired using a specially designed solution combining a CCD video camera with an NIR illumination system. The system is fully automatic and classifies rotation in all-view direction, detects eye blinking and eye closure and recovers the gaze of the eyes. Experimental results using real images demonstrates the accuracy and robustness of the proposed solution.

## 1 Introduction

The ever-increasing number of traffic accidents in the EC due to the diminished driver's vigilance level has become a serious problem to society. Driver fatigue resulting from sleep deprivation or sleep disorders is an important factor in the increasing number of accidents on today's roads. Statistics shows that a leading cause for fatal or injury-causing traffic accidents is due to drivers with a diminished vigilance level. Automatically detecting the visual attention level of drivers early enough to warn them about their lack of adequate visual attention due to fatigue may save a significant amount of lives and personal suffering. Therefore, it is important to explore the use of innovative technologies for solving the driver visual attention monitoring problem.

Many efforts have been reported in the literature on developing non-intrusive real-time image-based fatigue monitoring systems [2, 7–9, 11]. Measuring fatigue in the workplace is a complex process. There are four kinds of measures that are typically used in measuring fatigue: physiological, behavioral, subjective self-report and performance measures [15]. An important physiological measure that has been studied to detect fatigue has been eye-movements. Several eye-movements were used to measure fatigue like blink rate, blink duration, long closure rate, blink amplitude, saccade rate and peak saccade velocity. An increasing popular method of detecting the presence of fatigue is the use of a measure called PERCLOS [15]. This measure attempts to detect the percentage of eye-lid closure as a measure of real time fatigue. The present solution focuses

on rotation of the head and eye blinking, two important cues for determining driver visual attention, to gather statistics about the driver's visual attention level.

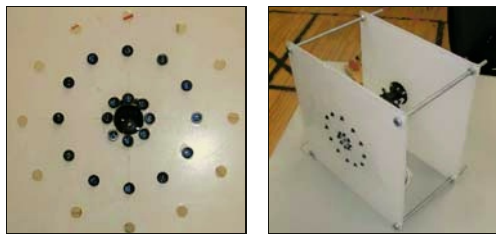
The organization of the paper is as follows. In section 2, the image acquisition system and illuminator is presented. The pupil detection solution based on the Purkinje images is presented on section 3. This entails pupil detection, tracking and eye gaze estimation. In section 4, the automated driver visual attention statistics and some results are given and in section 5 the details of the 3D head orientation and results are presented. Finally, conclusions are presented in section 6.

## 2 Image Acquisition System and Illuminator

To take advantage of the Purkinje images, a special camera-illuminator device was constructed. For that purpose, several NIR light emitting diodes (the TSHA650 from Vishay Telefunken) were distributed evenly and symmetrically along the circumference of two coplanar concentric rings [8] (see fig. 1). The center of the rings coincides with the camera optical axis. The IR light source illuminates the driver's eye and generates two kinds of pupil images: bright and dark pupil images. The bright pupil image is produced when the inner ring of IR leds is on and the dark pupil image when the outer ring is on. In order to take dark and bright pupil images simultaneously, the inner and outer ring control make use of the even/odd video signal information. The first Purkinje image, the so-called glint, is observed in both pupil images. A narrow band NIR filter (700-900 nm) was placed in front of the optical system of the camera to minimize interference from light sources beyond IR light and to maintain uniform illumination under different light conditions.

## 3 Pupil Detection, Tracking and Gaze Estimation

A robust and accurate pupil detection is crucial for the subsequent eyelid movements monitoring, eye gaze determination and face orientation estimation. Pupil detection is obtained by IR illumination after removing external illumination disturbance, and the result will be used on pupil tracking via Kalman filtering.



**Fig. 1.** Image Acquisition and NIR Illuminator.



**Fig. 2.** The bright and dark pupil effect.

### 3.1 Pupil and Glint Detection

At the NIR wavelength, pupils reflect almost all the IR light they receive along the path back to the camera, producing the bright pupil effect. If illuminated off the camera optical axis, the pupils appear dark since the reflected light will not enter the camera lens. This produces the so-called dark pupil effects.

Pupil detection involves locating pupils in the image. The narrow band NIR filter that was attached to the camera lens almost remove the ambient light interference. To robustly detect the pupils, each frame is separated into two image fields, representing the bright and dark pupil images separately (fig. 2). The image subtraction of these two image fields will produce an image with an high intensity contrast between the pupils and the rest of the image, allowing easy pupil segmentation via a simple global thresholding. This yields a binary image consisting of binary blobs that may represent the pupils. The pupils are detected by searching the entire image to locate two blobs that satisfy certain size, shape and distance constraints. The relationship between the shape and size of the pupils and the distance between each other is defined based on the anthropometric measures of the human face. After the correct detection of both pupils, an ellipse fitting is applied to each pupil and the centroid of the resulting ellipse is returned as the position of the detected pupil.

To take advantage of the high contrast between the glint and the rest of the image, the glint is detected using the dark image field. The bimodal intensity distribution of the dark image field allows a robust detection of the glint via simple image thresholding in the neighborhood region of the pupils. Once again, the shape and position distribution of the glints are used to constrain the segmentation results. Since the glints are visible in both image fields, the glints detected in the dark image field are cross-checked with the results obtained with the bright image field. The centroid of the segmented blob of a glint is returned as the image position of the glint.

### 3.2 Pupil Tracking

To continuously monitor the driver visual attention, it is important to track the eyes in real-time. We implemented a Kalman filter tracker to accomplish this task. This tracker is aimed to fulfill two purposes: estimate the position and

uncertainty of moving targets in the next frame and to filter out noise input data.

The target state vector is  $X = [p_l \ p_r \ g_l \ g_r \ \dot{p}_l \ \dot{p}_r \ \dot{g}_l \ \dot{g}_r]^T$  where  $p_i = (x_i, y_i)|_{i=p_r, p_l}$  and  $\dot{p}_i = (\dot{x}_i, \dot{y}_i)|_{i=p_r, p_l}$  are the image position and image velocity of the pupils and  $g_i = (x_i, y_i)|_{i=g_r, g_l}$  and  $\dot{g}_i = (\dot{x}_i, \dot{y}_i)|_{i=g_r, g_l}$  are the image position and image velocity of the glints.

The system model used is the following discrete model:

$$X_k = \mathbf{f}(X_{k-1}, k-1) + \mathbf{W}_k \quad Z_k = \mathbf{h}(X_k, k) + \mathbf{V}_k \quad (1)$$

where  $\mathbf{W}_k$  is a discrete-time white noise process with mean zero and covariance matrix  $Q$ ,  $\mathbf{V}_k$  is a discrete-time white noise process with mean zero and covariance matrix  $R$ , and  $\mathbf{W}_j$ ,  $\mathbf{V}_k$ , and  $X_0$  are uncorrelated for all  $j$  and  $k$ . We considered the assumption that trajectories are locally linear in 2D, resulting for the system model the following linear difference equation  $X_k = A \cdot X_{k-1} + W_k$  where the system evolution matrix,  $A_k$ , is based on first order Newtonian dynamics and assumed time invariant.

The measurement vector is  $Z_k = [p_l \ p_r \ g_l \ g_r]^T$  and is related to the state vector via the measurement equation  $Z_k = C \cdot X_k + \mathbf{V}_k$ .

The state covariance matrix  $P_k$  encodes the information of the ellipse of uncertainty of the estimation and can be used to compute the search area for the pupils and the glints. Specifically, the search area size was chosen as  $[H, W] = [20 + 0.2 \cdot P_k(y, y), 25 + 0.3 \cdot P_k(x, x)]$ .

### 3.3 Head-Eye Gaze Estimation

As stated before, the first and the fourth images of Purkinje (dual-images of Purkinje) supply a very reliable information for head-eye gaze estimation [3, 10]. When the head-eye is panned horizontally or vertically, the relative positioning of the glint and the centre of the bright-eye change accordingly, and the direction of gaze can be calculated from these relative positions.

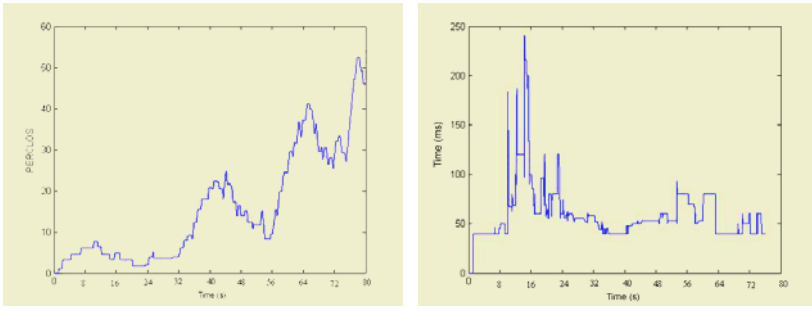
For a roll free head rotation, the locations of the pupils will share a common image line. In case of a pure roll head rotation (frontal orientation), the orientation of the line defined by both pupils gives an estimation of the roll angle of the head and the relative positioning of the glint and the pupil is the same in each one of the eyes. In the case of a head-eye yaw rotation, this relative positioning is different for each one of the eyes, being equal for the case of a pitch head-eye rotation. This observation is used to obtain a rough estimation of the direction of gaze.

Assuming roll free head rotation, the dual-images of Purkinje supply the following measures

$$D^{yaw} = (|x_{p_r} - x_{g_r}|) - (|x_{p_l} - x_{g_l}|) \quad D^{pitch} = 0.5 * ((y_{p_r} - y_{g_r}) + (y_{p_l} - y_{g_l})) \quad (2)$$

that are used to estimate the head-eye gaze orientation.  $D^{yaw}$  is null for a frontal head pose and shows positive/negative values for right/left head rotations. The eye gaze orientation is measured on the eye with less pupil-glint relative position.





**Fig. 3.** PERCLOS (left) and AECS (right) measurements over a period of 80 seconds.

Using these measures, the head-eye gaze orientation is obtained via a linear mapping procedure. To make these measures scale invariant, they are normalized by dividing over the inter-pupil distance value of the front view. An off-line calibration procedure was carried on, quantizing the head gaze orientation in steps of  $5^\circ$ .

## 4 Automated Driver Visual Attention Statistics

Of the drowsiness-detection measures, the measure referred to as PERCLOS was found to be the most reliable and valid determination of a driver's alertness level. PERCLOS is the percentage of eyelid closure of the pupil over time and reflects slow eyelid closures (droops) rather than blinks. To measure eyelid closure of the pupil, the size of the pupil was taken as the average size of both pupils and the rate of closure is defined as  $rate_{closure} = 1 - (pupilsize)/max(pupilsize)$ , defining a closed eye if  $rate_{closure} \geq 0.8$ .

AECS is the average eye closure speed [9], which means the amount of time needed to fully close the eyes and to fully open the eyes. An individual eye closure speed is defined as the time period during which the  $0.2 \leq rate_{closure} \leq 0.8$ . Figure 3 show the PERCLOS and AECS for a period of 80 seconds.

## 5 Driver Head Orientation

The presented approach models the shape of the driver's face with an ellipse, since human faces can be accurately modelled with an ellipse and is less sensitive to facial expression changes. To recover the 3D face pose from a single image, it is assumed that the ratio of the major and minor axes of the 3D face ellipse is known. This ratio is obtained through the anthropometric face statistics. Our purpose is to recover the three angles of rotation: yaw (around vertical axis), pitch (around horizontal axis) and roll (around the optical axis).

### 5.1 Image Face Ellipse Detection and Tracking

The image face ellipse detection and tracking is based on three major steps: i) obtain an approximate location of the face based on the positions of the eyes.



**Fig. 4.** Image face ellipse detection.

Since the pupil position varies as a function of the eye gaze movements, the approximate location of the face is based on the location of the glints which are invariants to the eye gaze. ii) determine the best fitted ellipse for the image face by maximizing the normalized sum of the gradients around the edges of the face. iii) Ellipse face tracking using a Kalman filter.

In order to correctly detect the face ellipse, some constraints must be considered, in special size, location and orientation. The distance between the detected glints and their location are used to constrain the size and location of the image face ellipse. The orientation of the line that passes through both glints is directly related to the 3D face roll rotation. For roll free face poses this line remains horizontal, which means that it is invariant to the yaw and pitch rotations. Under this constraints, the roll angle ( $\psi$ ) is defined by  $\psi = \text{atan}[(y_{p_l} - y_{p_r}) / (x_{p_l} - x_{p_r})]$ .

Under frontal orientation, a weak perspective projection can be assumed and the face symmetry for the location of the eyes within the 3D face ellipse hold for the image face ellipse. This means that the major axis of the face ellipse is normal to the line connecting the two glints and pass through the center of the line. In fact, these constraints doesn't hold for non-frontal orientation and the orientation of the major line is not normal to the connecting line. Although, the solution adopted kept the constrain that the major axis of the ellipse pass through the center of the line, considering the existence of an angle  $\alpha$  between the major axis and the normal to the line that connect the two glints.

Assuming the existence of an ellipse coordinate frame located at the middle point of the glints connecting line, with the  $X$  and  $Y$  axis aligned with the minor and major axes of the ellipse, respectively, the image face ellipse is characterized by four parameters  $(m_i, n_i, d, \alpha)$ , where  $m_i$  and  $n_i$  are the lengths of the major and minor semi-axis of the ellipse, respectively,  $d$  is the distance to the image ellipse center and  $\alpha$  is the rotation angle.

Taking the approach proposed by Birchfield [4], the image face ellipse can be detected as the one that minimizes the normalized sum of the gradient magnitude projected along the directions orthogonal to the ellipse around the perimeter of the ellipse. This can be formulated has  $\varepsilon = \frac{1}{N} \sum_{i=1}^N |n(i) \cdot g(i)|^2$  where  $n(i)$  is the unit vector normal to the ellipse at pixel  $i$ ,  $g(i)$  is the pixel intensity gradient and  $(\cdot)$  denotes dot product. The best face ellipse is  $\chi = \text{arg max}_{e \in E} (\varepsilon^2)$  where the search space  $E$  is the set of possible ellipses produced by varying the four parameters of the ellipse. In order to constraint the searching space, the rough estimation of the 3D face orientation obtained via the dual-images of

Purkinje is used to define an initial estimate for these parameters. The four ellipse parameters are tracked via a kalman filter. Figure 4 show the result of the image face ellipse detection.

### 5.2 Face Orientation

Consider an object coordinate frame attached to the 3D face ellipse, with its origin located on the center of the ellipse and its  $X$  and  $Y$  axes aligned with the major and minor axes of the ellipse. The  $Z$  axis is located normal to the 3D ellipse plane. The camera coordinate frame is located at the camera optical center with the  $X_c$  and  $Y_c$  aligned with the image directions with the  $Z_c$  along the optical axis. Since the 3D face ellipse is located on the plane  $Z = 0$ , the projection equation that characterizes the relationship between an image face ellipse point  $p_i = (x, y, 1)^T$  and the corresponding 3D face ellipse point  $P_i = (X, Y, 1)^T$  is given by  $p_i = \beta K[R|t]P_i$  where  $K$  represents the camera intrinsic parameters matrix,  $M = [R|t] = [r_1 \ r_2|t]$  is the extrinsic parameters matrix and  $\beta = \lambda/f$  is an unknown scalar.

Representing

$$[x \ y \ 1] \begin{bmatrix} a & c/2 & d/2 \\ c/2 & b & e/2 \\ d/2 & e/2 & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \tag{3}$$

the matricial generic formula of an ellipse, the 3D face ellipse and the image face ellipse can be defined, respectively, as

$$[X \ Y \ 1] \mathbf{Q} [X \ Y \ 1]^T = 0 \quad [x \ y \ 1] \mathbf{A} [x \ y \ 1]^T = 0. \tag{4}$$

Substituting  $p_i = \beta KMP_i$  to Eq. 4 lead to

$$[X \ Y \ 1] \beta M^T K^T A K M [X \ Y \ 1]^T = 0. \tag{5}$$

Denoting  $B = K^T A K$ , the 3D ellipse matrix  $Q$  yields  $Q = \beta M^T B M$ .

Let the length of the major and minor axis of the 3D face ellipse be  $m$  and  $n$ , respectively, and since the object frame is located on the center of the ellipse, the ellipse matrix  $Q$  is parameterized as

$$Q = \begin{bmatrix} 1/m^2 & 0 & 0 \\ 0 & 1/n^2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{6}$$

resulting the equation

$$\begin{bmatrix} 1/m^2 & 0 & 0 \\ 0 & 1/n^2 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \beta \begin{bmatrix} r_1^T B r_1 & r_1^T B r_2 & r_1^T B t \\ r_2^T B r_1 & r_2^T B r_2 & r_2^T B t \\ t^T B r_1 & t^T B r_2 & t^T B t \end{bmatrix} \tag{7}$$

Due to the symmetry of the matrix, there are only six equations (constraints) for a total of nine unknowns.



Fig. 5. Head face orientation estimation.

Since the roll angle was already obtained, the face orientation can be defined just by the yaw and pitch rotation. Assuming a null translation vector, the rotation matrix obtained from the yaw and pitch rotation is

$$R = R_{\sigma}R_v = [r_1 \ r_2 \ r_3] = \begin{bmatrix} \cos(\sigma) & \sin(\sigma)\sin(v) & -\sin(\sigma)\cos(v) \\ 0 & \cos(v) & \sin(v) \\ \sin(\sigma) & -\cos(\sigma)\sin(v) & \cos(\sigma)\cos(v) \end{bmatrix}. \quad (8)$$

Assuming that the ratio between the major and minor axis if the 3D face ellipse is know by anthropometric face analysis, and letting  $c = m^2/n^2$  represent this ratio, the  $2 \times 2$  sub-matrix yields

$$\beta \begin{bmatrix} r_1^T B r_1 & r_1^T B r_2 \\ r_2^T B r_1 & r_2^T B r_2 \end{bmatrix} = \begin{bmatrix} 1/m^2 & 0 \\ 0 & 1/n^2 \end{bmatrix} \quad (9)$$

resulting the following constraint equations

$$r_1^T B r_2 = 0 \quad (10)$$

$$\frac{\beta r_1^T B r_1}{1/m^2} = \frac{\beta r_2^T B r_2}{1/n^2} \Leftrightarrow r_1^T B r_1 = \frac{n^2}{m^2} r_2^T B r_2 \Leftrightarrow r_2^T B r_2 - c r_1^T B r_1 = 0. \quad (11)$$

Using these two equations it is possible to solve for the pitch and yaw iteratively. The initial estimates of  $0^\circ$  for both angles has been used with correct convergence results. This approach was tested with several real images with good results. Although, the accuracy obtained with this approach is highly dependent on the image face ellipse obtained. Figure 5 show the results obtained with the face orientation estimation approach.

## 6 Conclusions

A Real-time Driver Visual Attention Monitoring System was presented. A special hardware image acquisition and illuminator system was described to take advantage of the dual-images of Purkinje. A efficient and simple solution for pupil detection was presented that were used to take some drossiness measure in real-time. A rough estimation of the head-eye gaze was described based on the dual-images of Purkinje and finally an ellipse based face orientation estimation

was presented. Although the good results obtained with the face orientation estimation, it reveals to be highly dependent on the image face ellipse detection. Further research is necessary in order to improve the accuracy of the image face ellipse detection.

## References

1. H.D. Crane, The Purkinje Image Eyetracker, Image Stabilization, and Related Forms of Stimulus Manipulation, *In Visual Science and Engineering: Models and Applications*, D. H. Kelly, Ed. Marcel Dekker, Inc., New York, NY, 1994.
2. A. Yilmaz, et al., Automatic feature detection and pose recovery for faces. *ACCV2002*, Melbourne, Australia, 2002.
3. T. N. Cornsweet et al., Accurate two-dimensional eye tracker using first and fourth purkinje images, *J. Opt. Soc. Amer.*, vol. 63, no. 8 August 1973.
4. S. Birchfield, Elliptical head tracking using intensity gradients and color histograms, *IEEE CVPR*, 1998.
5. A.H. Gee et al., Determining the gaze of faces in images, *Image and Vision Computing*, 12 (10), 1994.
6. A.T. Horprasert et al., Computing 3D head orientation from a monocular image sequence *SPIE, 25th AIPR workshop: Emerging Applications of Computer Vision* 2962, 1996.
7. P. Smith et al., Determining Driver Visual Attention with one camera, *IEEE Trans. on Intelligent Transportation Systems*, vol.4, no.4, Dezember, 2003.
8. Qiang Ji et al., Real-time eye, gaze and face pose tracking for monitoring driver vigilance, *Real-Time Imaging*, 8, 2002.
9. Qiang Ji et al., 3D pose estimation and tracking from a monocular camera, *Image and Vision Computing*, 2002.
10. J.G. Wang et al. Study on Eye Gaze Estimation, *IEEE Trans. on Systems, Man, and Cybernetics, PART B: Cybernetics*, vol. 32, no. 3, JUNE 2002.
11. R. Grace, A drowsy driver detection system for heavy vehicles, *Conf. on Ocular Measures of Driver Alertness*, 1999.
12. C. Morimoto et al., Pupil detection and tracking using multiple light sources, *Image and Vision Computing*, 18, 2000.
13. S.Y. Ho et al., An analytic solution for the pose determination of human faces from a monocular image, *In Pattern Recognition Letters*, 19, 1998.
14. M. Mallis et al., Ocular measurement as an index of fatigue and as the basis for alertness management: experiment on performance-based validation of technologies, 1999.
15. P. Sherry et al., Fatigue Countermeasures in the Railroad Industry: Past and Current Developments, *Published by Association of American Railroads*, 2000.

# An Approach to Vision-Based Person Detection in Robotic Applications

Carlos Castillo and Carolina Chang

Grupo de Inteligencia Artificial  
Universidad Simón Bolívar  
Caracas 1080, Venezuela  
carlos@gia.usb.ve, cchang@ldc.usb.ve

**Abstract.** We present an approach to vision-based person detection in robotic applications that integrates top down template matching with bottom up classifiers. We detect components of the human silhouette, such as torso and legs; this approach provides greater invariance than monolithic methods to the wide variety of poses a person can be in. We detect borders on each image, then apply a distance transform, and then match templates at different scales. This matching process generates a focus of attention (candidate people) that are later confirmed using a trained Support Vector Machine (SVM) classifier. Our results show that this method is both fast and precise and directly applicable in robotic architectures.

## 1 Introduction

Detection and recognition of objects from images disregarding orientation, scale and view is a very important research subject in computer vision. People detection in images and video sequences is a research subject in this area. We are interested in this problem from a robotic application point of view since we are currently in early development stages of a robotic application for search and rescue operations [2].

The problem of people detection is very complex and has not been solved in its generality, but there have been advances where the pose is fixed, such as in the case of pedestrians [1, 9, 14]. However not much attention has been given to the problem when the camera cannot be assumed stationary (therefore not having a explicit scene model).

Our approach uses fast template matching as a focus of attention. Basically it discards locations where there is no silhouette matching the human body. And from those candidate locations (ideally, a very reduced set), we query a full scale SVM.

The contributions pretended are two-fold: first the design and implementation of a vision system that integrates top-down template matching with bottom-up classifiers; and second a concrete implementation on board a robot in an embedded application.

The rest of the paper is organized as follows, first we describe distance transforms for template matching and then support vector machines for pattern recognition, after that we describe the system details, then the results are presented. Finally, the discussion and conclusions are presented and then ideas for future work are given.

## 2 Distance Transform for Template Matching

A distance transform (DT) converts a binary image (containing values 0 and  $\infty$ ) to an image where each pixel value denotes the distance to the nearest feature pixel. From this definition of the distance transform problem, a  $O(n^4)$  algorithm can be readily constructed (for an  $n \times n$  image). However, over the last 20 years the state of the art has advanced either approximating the EDT in a  $O(n^2)$  time or providing an exact solution in a  $O(n^3)$  time.

Many DT algorithms exist, the differing characteristic is the distance metric and the propagation of local distances. In particular we use Euclidean distance and Maurer's line-column scanning method [10].

After the image has been adequately preprocessed the template matching step begins. As described in by Gavrilu [7], a given image  $I$  is said to be matching a template  $T$  when:

$$D(T, I) \leq \theta \quad (1)$$

where  $\theta$  is a user defined threshold on the maximum acceptable dissimilarity between the DT image and the template, and  $D(T, I)$  is given by:

$$D(T, I) = \frac{1}{|T|} \sum_{t \in T} d_I(t) \quad (2)$$

where  $|T|$  is the number of features in  $T$  and  $d_I(t)$  is the distance between feature  $t \in T$  and the closest feature in  $I$ .

## 3 Support Vector Machines for Pattern Classification

Support vector machines (SVMs) is a principled machine learning technique that is well founded in statistical learning theory.

SVMs have two outstanding characteristics: (1) they have a solid mathematical foundation and (2) strong practical results in large-scale, real-world problems.

Traditional machine learning methods such as backpropagation, minimize the training error, while SVMs minimize a bound on the empirical error and the complexity of the classifier, simultaneously. Therefore, SVMs are likely to perform better than conventional techniques, such as backpropagation trained neural networks. The decision surface of an SVM is given by:

$$f(x) = \text{sgn} \left( \sum_{i=0}^{N_s} \alpha_i y_i K(x, x_i) + b \right) \quad (3)$$

where  $N_s$  is the number of support vectors (points closest to the separating hyperplane, in terms of which the decision boundary is defined);  $x$  is the point to be classified,  $x_i$  is a support vector, and  $\alpha_i$  is the corresponding Lagrangian multiplier.  $K$  is a kernel satisfying Mercer's conditions. For a complete review of SVMs for pattern recognition (see [4]).

## 4 System Details

At its core, our system for person detection uses template matching employing Euclidean distance transform (EDT) to evaluate candidate people by independent components (such as torso, leg, arm, head). These matched components are immediately verified using a SVM specialized for that component. If valid, the component is adequately marked on the image. The very first step is preprocessing. Each input image is grayscaled and contour-filtered using the Marr-Hildreth method[11]. After that, the contoured and grayscaled (CG) image is transformed using an EDT. Figure 1 shows the result of running the preprocessing step on three example images.

We have devised two simple methods for image scanning:

- Using exhaustive scanning. In an  $X \times Y$  image with an  $N \times M$  template, we first try to match the window defined by the rectangle  $(0, 0, N, M)$ ; after that the one defined by  $(1, 0, N + 1, M)$ , and so on until reaching the end of the image at that scale.
- Using random sampling. In an  $X \times Y$  image with an  $N \times M$  template, we select a fixed number of samples proportional to the size of the image. This scanning method accelerates the process with a sacrifice in precision.

In the offline experiments we use exhaustive scanning because runtime performance is not an issue. However, the online version uses the randomized method.

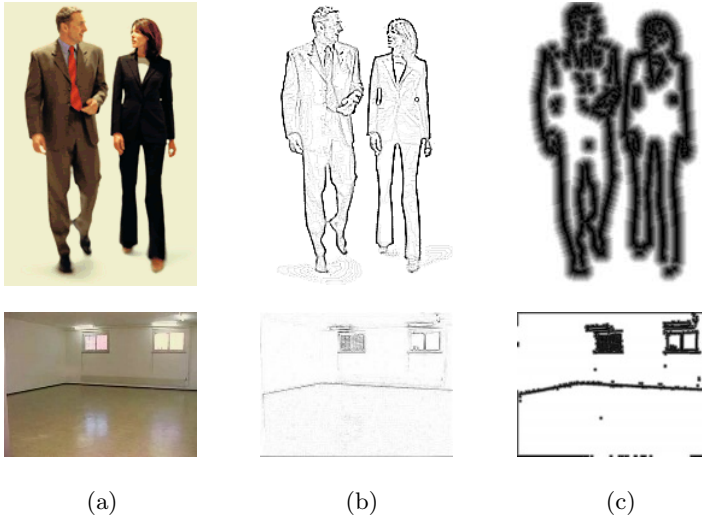
After experimentation we settled with 12 templates. More templates means a better definition of the class of interest but also translates into a slower matching process. The templates are taken from photographs of the object of interest after contour filtering it and obtaining the relevant connected components.

When an image window matches a template, a previously trained and bootstrapped SVM is queried. If the SVM classifies the window as a valid component, the component is then marked in the original image taking into account the scale. Compared to template matching, SVM query phase is very slow. We have looked into simplifying the verification and use Burges's method [3] but later noticed that a homogenous quadratic kernel does not perform well on some of these component datasets.

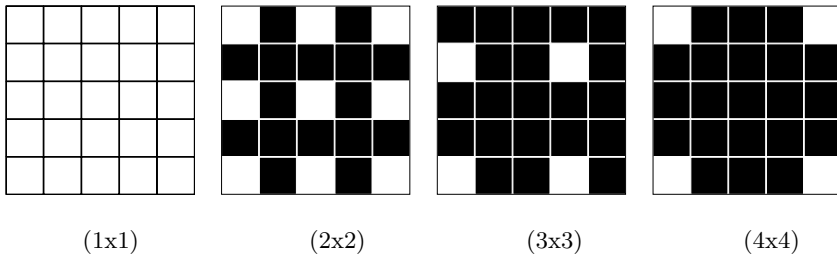
This approach is not new. Heisele et al. [9] and Gavrila [7], both use some type of hierarchical quick discard method. However, our method is very simple and uses a small amount of templates compared to the results reported by Gavrila [7].

The initial prototype of the current system was written in Python. It uses the LIBSVM support vector machine library [5]. For image processing, we used





**Fig. 1.** (a) is the original image, (b) is the contoured and grayscale image and (c) is the distance transformed image ready for template matching.

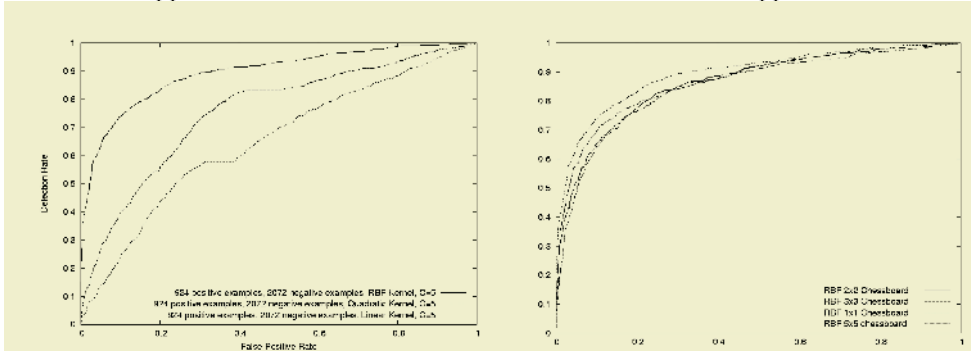


**Fig. 2.** Chessboard feature selection for various sizes. White squares represent selected pixels, black squares represent non-selected pixels.

the Python Imaging Library (PIL). The production version of the system is written in C++ and uses LIBSVM and ImageMagick. The main difference in the two implementations is mainly performance. On our 1.6 GHz Pentium IV machine, the C++ version runs at 3 frames per second. The system does not use movement as a focus of attention; using movement our system should be considerably faster.

## 5 Results

We use a chessboard sampling of the pixels in the input image, as presented in Fig. 2. The ROC (Receiver Operating Characteristic) curves in Fig. 3 (right) show that the loss in accuracy is not significant while this feature selection method makes real time performance feasible for our approach. The fact that



**Fig. 3.** Left: ROC Graphic of the SVM classifier with an RBF (Radial Basis Function), quadratic and linear kernel and  $C=5$ . Both classifiers are of similar complexity. Notice the poor performance of the linear and quadratic kernels. Right: ROC Graphic of the SVM classifier with an RBF (Radial Basis Function) and different chessboard intervals. The loss in accuracy can also be observed in Table 1.

**Table 1.** Chessboard feature: selection and mean and standard deviation of the classification rate doing a 5-piece cross-validation of the torso classifier.

Features	Mean $\pm$ Std. Dev.
1x1	89% $\pm$ 1%
2x2	86% $\pm$ 1%
3x3	86% $\pm$ 1%
5x5	87% $\pm$ 1%
7x7	83.5% $\pm$ 1%

this type of very simple feature selection approach works shows that the training data are highly redundant.

We applied a 5-piece cross-validation of the training set and report the mean and standard deviation of the classification accuracy rate of the torso classifier in Table 1. Results show that we obtained high accuracy rates on a very large complex dataset.

In Fig. 3 (left) it can be clearly observed that the linear and quadratic kernel perform very poorly in this domain. While using a quadratic kernel, Burges’s method [3] can be readily applied, as reported by Papageorgiou and Poggio [13] after results reported by Osuna et al. [12] in another domain. We consider the precision loss to make this approach prohibitive.

We present several examples of the output of the offline version of the system in Fig. 4. Notice that kids are detected by the system. We consider this to be encouraging since their characteristic proportions are different to those of an adult. The system is also able to correctly classify a naked torso. This is remarkable since the torso of a naked person is considerably different to the torso of a dressed person.



**Fig. 4.** The first two rows contains examples of the system running on several images offline. The last row shows results obtained by the online version of the system in our office environment.

## 6 Robotic Application

We tested the system onboard an ActivMedia Robotics Pioneer 2 mobile robot. The online version (onboard the robot) uses the randomized scanning method previously described.

It is important to note that because the camera is not stationary and the background is constantly varying, simple techniques of background subtraction cannot be used for getting the foreground objects. We execute multi-scale exhaustive scanning at each frame.

Because a robotic application usually needs to be run on hardware that is not last generation, we found the querying an SVM on every candidate quickly becomes a crippling bottleneck. We eliminated the SVM querying step from the online version.

The performance (as measured by false positives and false negatives) degenerated significantly. To handle this we adjusted (downwards) the value of  $\Theta$  in the template matching step. Further, to enhance the precision of the system in our office environment, we measured the correlation of the value pixels on the DT image over the template as described in equation 2 and called this value  $\beta$  and measured the percentage of matching non-data points in the template compared to the contoured image and called this value  $\alpha$ . So the matching criteria is:

$$\frac{\alpha}{\beta} > \gamma \quad (4)$$

where  $\gamma$  is an experimentally set threshold value. The matching criteria seeks a balance of many matched points with low matching error (derived from the distance measure of the EDT image). This refinement of the matching criteria significantly decreases the false-positive rate and eliminates the need of querying an SVM to have acceptable results.

The online version of the system works at 3 Hz.

## 7 Conclusions

We have presented an approach to vision-based person detection in robotic applications that integrates top down (high speed) template matching with bottom up classifiers. We detect components of the human silhouette such as torso and legs; this approach provides greater invariance than monolithic methods to the wide variety of poses a person can be in.

The torso detection methodology presented currently works very well even though each pattern contains more than 1400 features. We have found that the torso can be characterized as very noisy data due to the presence of clothes. The trained SVM classifier correctly captures the relevant information to classify a torso from CG image data, yet querying it is a bottleneck that makes unfeasible to run the system in real time. We presented an alternative using only template matching.

We believe this shows the wide range of applicability of our approach. Our torso dataset contains 924 torsos (from the MIT Pedestrian dataset) and 2072 non-torsos (the non-torsos were generated after a bootstrapping process).

Developing classifiers and templates for other components of the human body (more important in other poses) for use by this method constitutes promising future work. By detecting components of the human body our method is more resilient to occlusion than monolithic approaches.

Our system is not ready for mission critical applications. Performing a principal component analysis (instead of the described chessboard) for feature selection would be a challenging future direction with this large-scale dataset. In the future, we intend to automatically construct shape models using techniques such as described by Duta et al. [6] and Gavrilu et al. [8] to generate a larger template set before continuing on to the development of the classifiers for search and rescue poses.

## References

1. M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi. Shape-based pedestrian detection and localization. *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems*, pages 328–333, 2003.
2. A. Brando and C. Chang. Firefighter-robot interaction during a hazardous materials incident exercise. In *11th International Conference on Advanced Robotics*, volume 2, pages 658–663, 2003.

3. C. J. C. Burges. Simplified support vector decision rules. In *International Conference on Machine Learning*, pages 71–77, 1996.
4. C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
5. C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
6. N. Duta, A. K. Jain, and M. P. Dubuisson-Jolly. Automatic construction of 2d shape models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):433–446, 2001.
7. D. Gavrila. Pedestrian detection from a moving vehicle. *Proc. of the European Conference on Computer Vision*, 2(8), 2000.
8. D. Gavrila, J. Giebel, and H. Neumann. Learning shape models from examples. In *Pattern Recognition, 23rd DAGM-Symposium, Munich, Germany, September 12-14, 2001, Proceedings*, volume 2191 of *Lecture Notes in Computer Science*. Springer, 2001.
9. B. Heisele, C. Nakajima, M. Pontil, and T. Poggio. People recognition in image sequences by supervised learning. Technical Report CBCL-188, MIT Artificial Intelligence Laboratory, June 7 2000.
10. C.R. Maurer Jr. and V. Raghavan. A linear time algorithm for computing the euclidean distance transform in arbitrary dimensions. In *IPMI*, 2001.
11. D. Marr and E. Hildreth. Theory of edge detection. *Proc Roy. Soc. London*, page B207:187, 1980.
12. E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), June 17-19, 1997, San Juan, Puerto Rico*. IEEE Computer Society, 1997.
13. C. Papageorgiou and T. Poggio. Trainable Pedestrian Detection. In *Proceedings of the 1999 International Conference on Image Processing (ICIP-99)*, pages 35–39, Los Alamitos, CA, October 24–28 1999. IEEE.
14. C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

# A New Approach to the Template Update Problem<sup>\*</sup>

Cayetano Guerra, Mario Hernández,  
Antonio Domínguez, and Daniel Hernández

IUSIANI - Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas  
en Ingeniería, Univ. de Las Palmas de Gran Canaria

35017, Campus de Tafira

Las Palmas de Gran Canaria, Spain

{cguerra,mhernandez,acdbrito,dhernandez}@iusiani.ulpgc.es

**Abstract.** Visual tracking based on pattern matching is a very used computer vision technique in a wide range of applications [4]. Updating the template of reference is a crucial aspect for a correct working of this kind of algorithms. This paper proposes a new approach to the updating problem in order to achieve a better performance and robustness of tracking. This is carried out using a representation technique based on second order isomorphisms. The proposed technique has been compared experimentally with other existing approaches with excellent results. The most important improvements of this approach is its parameter-free working, therefore no parameters have to be set up manually in order to tune the process. Besides, objects to be tracked can be rigid or deformable, the system is adapted automatic and robustly to any situation.

## 1 Introduction

Visual tracking based on pattern matching is a very used computer vision technique in a wide range of applications [4]. Its working is simple, a template of reference is searched in the current image. However, updating the template of reference is a crucial aspect since the object of interest normally modifies its visual aspect through the time. Therefore, an adaptation of the pattern is necessary in order to keep the object.

Two problems can arise due to the insufficient or excessive frequency of the number of updates. In the first of them, the visual aspect of the object of interest can become too different from the pattern and, in this way, the searching algorithm can find other part of the searching window more similar to the current pattern. This produces a *jump* in the object of interest. The other problem is due to applying too and unnecessary updates to the tracking process. The digital

---

<sup>\*</sup> This work has been partially supported by the Spanish Ministry of Education and Science and FEDER funds under research project TIN2004-07087, the Canary Islands Regional Government under projects PI2003/165 and PI2003/160 and the University of Las Palmas under projects UNI2003/10, UNI2004/10 and UNI2004/25.

nature of images and patterns can cause *drifting* due to an accumulative sub-pixel error in every update. Sometimes, the random movement of the object can counteract the effect of the drift, but with certain kind of movements the drift can be significant and end up losing the object. Every update causes a potential drift.

This work proposes a new template updating approach within the framework of representation spaces based on second order isomorphisms. Among its advantages are a parameter free working, no parameter have to be set up manually prior working, and a better performance than the traditional updating methods.

## 2 Second-Order Isomorphisms

The "objects" are located in the real world and, after Shepard [11], we will name to this world *Distal Space*. Every object in this space will have its own representation in an inner space  $\Phi$ , named *Proximal Space*. In this work we define *Visual Object* to any physical entity in the real world which has associated its own internal representation. In the proximal space the goal of the visual system is to assign to every visual object in the distal space a unique symbol in a proximal space, and thereby to establish an isomorphism between both spaces, [6].

Besides this correspondence, it is even much more useful to establish relations among objects in a distal space and their respective representations in the proximal space. A *second order isomorphism* [7, 11] should accomplish that if similarity between two distal objects  $A$  and  $B$  is greater than between distal objects  $B$  and  $C$ , then the distance between their respective representations ( $A'$ ,  $B'$  and  $C'$ ) should verify that  $d(A', B') < d(B', C')$ . Therefore, the representation schema not only stores information about the objects but also information about their relationships.

## 3 View-Based Representation Spaces

View-based approaches have experienced a renewed interest in the computer vision community in the last decade. After Bergen and Adelson [1], the appearance of a visual object in terms of images is described by the plenoptic function. That is, if the plenoptic function of a visual object is known, then every possible view of that object can be generated. This function depends on a set of parameters, like viewing position and lighting conditions, whose variability defines the appearances subspace corresponding to the visual object [3] in the views space.

This function was originally defined for rigid objects. However, if time varying parameters are included among the set of parameters  $\rho(t)$ , the plenoptic function  $V((\mathbf{x}), \rho(t))$  will be able of dealing with non-rigid visual objects. We can call to this function *generalized plenoptic function*. Unfortunately, finding the plenoptic function corresponding to an object in a certain scene is a very complex problem.

In order to overcome this drawback much effort has been done in the study of the views space. To characterize precisely the variability of images and other perceptual stimuli, a mathematical approach can be taken.

The views space can be modelled in image coordinates, based on considering the set of  $n \times m$  pixels corresponding to each image as a  $R^{m \times n}$  vector. We can consider each image as a vector with dimension  $m \times n$ . The set of all possible images of any distal object is a continuous subset of the views space [10]. This continuity is related to the smooth variation of visual aspect with respect to the plenoptic parameters. This can be stated as a *continuity principle* in the following manner: given an arbitrarily small  $\tau$  and  $\delta_d$ , the following condition will comply:

$$d[V(\mathbf{x}; \boldsymbol{\rho}(t)), V(\mathbf{x}; \boldsymbol{\rho}(t + \tau))] \leq \delta_d, \forall \mathbf{x} \in \mathbf{S} \quad (1)$$

Where  $\mathbf{S}$  corresponds to the support set of  $\mathbf{V}$  and  $d$  is a defined distance function. Varying  $t$ , in the generalized case, the set of points corresponding to the images of a distal object are in a manifold [10]  $\mathcal{M}_x$  of the Views Space. The manifold of a certain object  $O$   $\mathcal{M}_x^O$  is a lower dimensional subspace embedded [3, 5] in the views space with the  $l$  parameters of the plenoptic function as intrinsic dimensions:

$$\mathcal{M}_x^O = \{V(\mathbf{x}; \boldsymbol{\rho}(t)) \mid \boldsymbol{\rho} \in R^l\} \quad (2)$$

During a tracking process of an object, this does not show all possible views of itself included in its manifold but just a subset of them. This manifold subset,  $I(\mathbf{x}; t) \subset \mathcal{M}_x^O$ , will shape as a parametric curve of the time. We name this curve *Visual Transformation Curve of the Object*.

The tracking process tries to follow the visual object through this curve obtaining the values  $\alpha_0$  corresponding to the location of the best match at time  $t$ , through a function like:

$$\alpha_0(t) = \arg \min_{\alpha} \sum_x d(W(I(\mathbf{x}; t); \alpha), T(\mathbf{x}; t)) \quad (3)$$

Where  $I(.;.)$  is the image where looking for the template  $T(.;.)$  by means of a windowing function  $W(.;.)$ , which extracts an area of the same size than  $T(.;.)$  at position  $\alpha$ .  $\alpha_0(t)$  will be the minimum of the matching function, valued over all possible values of  $\alpha$ . That is, position of the window  $W$  over the image  $V$ .

The template tracking depends on the definition of several elements. Once defined the matching strategy and distance function to be used, the fundamental element to be defined is the template update strategy or, in other words, the steps in which the visual transformation curve is tracked.

## 4 Existing Updating Techniques

A number of strategies have been proposed to define the template to be used during the tracking process. In [2, 9] there are good surveys about these techniques. Strategies goes from no template updating at all, others with very naive approaches and some of them using similarity thresholds.

Among them, *template update based on statistics* [8] tries to overcome the inherent problems of drifting and jumps of interest seeking a balance in the number of updates to perform. This updating schema takes into account that

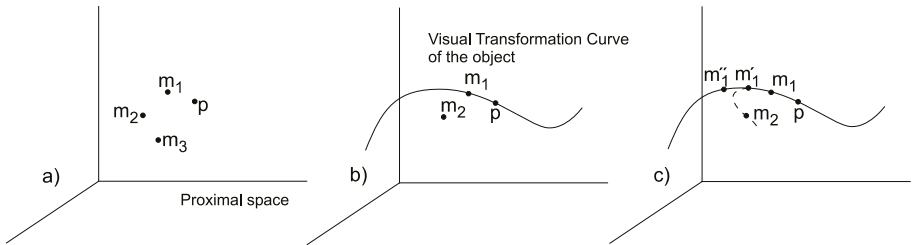


exceeding a similarity threshold provides only specific information and does not provide information at all about the rest of the image.

However, the quality of a maximum (or minimum) relies on the values that surround it. Therefore, this statistical method of updating considers the rest of values of the similarity function, in such a way that if the maximum (or minimum) is differentiated enough from the rest of values then the quality of the current pattern is good. This level of differentiation is calculated based on a statistical function, [8].

## 5 Proposed Template Matching Updating Technique

In order to describe the procedure proposed in this paper, we will denote as  $\mathbf{p}$  to a point corresponding to the representation of a visual object in the proximal space  $\Phi$  at a certain time, i.e.  $\mathbf{p}$  will correspond with the template  $T$  (see expression 3) in the space defined by  $R^{m \times n}$ . A distance  $d$  can be established in  $\Phi$ . In this work, the distance  $d$  between two points  $\mathbf{p}_1$  and  $\mathbf{p}_2$  is based on the  $L_2$  norm. This distance will be used between input image and template in order to obtain the best match.



**Fig. 1.** The figure depicts the points corresponding to the symbolic representations of the different searches over an image in a tracking process. The diagram c) illustrates the consequences of a lack of required updating.

After applying the distance function between image and pattern sliding the template over the searching window according to expression 3, a variable number of local minima will show up, among them, the absolute one. In  $\Phi$ , see figure 1, a) the vector  $\mathbf{p}$  corresponds to the pattern of reference, i. e. the view of the object of interest to look for. The vector  $\mathbf{m}_1$  will be the absolute minimum since it is the visual object most similar to the object of interest. The existence of more local minima,  $\mathbf{m}_2$  and  $\mathbf{m}_3$ , implies that there are other similar objects in a certain degree to the object of interest. We name them *objects of the context*. These objects, like the object of interest, have also their own curve of visual transformation included in their manifolds of the proximal space. Although for the sake of simplicity these objects of the context will remain static, see figure 1, b). The *Visual Transformation Curve* of the object of interest is the loci of the

points corresponding to the different minima after the matching process on input images during a certain time. This curve will be composed by the nearest vectors ( $\mathbf{m}_1, \mathbf{m}'_1, \mathbf{m}''_1, \dots$ ) to the pattern of reference ( $\mathbf{p}$ ). Therefore,  $\mathbf{m}_1$  corresponds to the closest point to  $\mathbf{p}$  in the moment  $t = 0$ ,  $\mathbf{m}'_1$  corresponds to the closest point to  $\mathbf{p}$  in the time  $t = 1$  and so on. However, if there exist, at least, one object of the context,  $\mathbf{m}_2$ , and the pattern of reference ( $\mathbf{p}$ ) is not updated, it may occur that, after a number of frames, the absolute minimum does not correspond to the real object of interest but to the most similar object of the context, as figure 1, c) shows. Thus, the area of the searching window corresponding to the point  $\mathbf{m}_2$  will be taken as the object of interest, resulting in an error of the tracking process, that is an *interest jump error*, which is a very common error of updating techniques that do not update the pattern just in time.

The origin of the problem is caused by the lack of updating or an inappropriate updating rate of the pattern of reference. It can be seen in figure 1, c) that  $d(\mathbf{p}, \mathbf{m}_1) < d(\mathbf{p}, \mathbf{m}_2)$  and  $d(\mathbf{p}, \mathbf{m}'_1) < d(\mathbf{p}, \mathbf{m}_2)$  but  $d(\mathbf{p}, \mathbf{m}'_1) > d(\mathbf{p}, \mathbf{m}_2)$ . For the sake of clarity the most similar object of the context,  $\mathbf{m}_2$ , does not move and consequently does not draw any visual transformation curve.

It is clear that the pattern should be updated before any object of the context can be more similar to the pattern of reference than the current view of the object of interest. To accomplish this an updating threshold must be set up taking into account the closeness of the *objects of the context*. Therefore, when a new view of the object of interest is taken as current pattern a new updating threshold is also computed automatically. The assigned value can be obtained by the rule of dividing by two the distance to the closest object of the context to the new pattern.

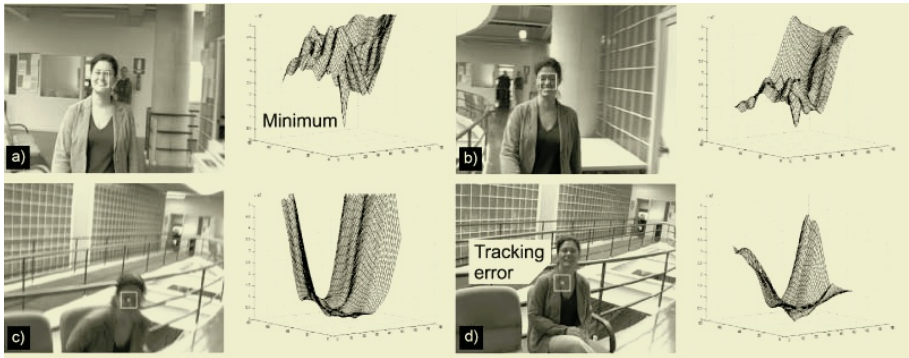
## 6 Experimental Results

Pattern updating is necessary if the view of the object of interest changes through the time. Besides, this updating must be done at the right moment in order to avoid the two most significant errors in a tracking process: *drifts* and *interest jumps*. These two kinds of error will mark experimentally the goodness of the different updating approaches.

Several experiments have been done in order to evaluate the performance of the proposed solution. Among them, two critical sequences, described in this paper, demonstrate the higher level of robustness of the new approach in comparison with the existing updating methods. Actually, only statistics updating based method [8] is used as the other methods are too simple and their limitations are obvious.

A complete tracking module has been developed to carry out the presented experiments. To obtain the results only the updating schema of this algorithm has been changed. In order to evaluate the updating approaches, the best method will be the one that carries out a correct tracking (without interest jump nor drifting errors) with the smallest number of updates.

In the first experimental sequence, see figure 2, a person walks and her face is tracked. At first sight, it seems a not problematic task. However, an error



**Fig. 2.** Different frames of a sequence where a face is tracked. Sometimes, at first sight and if similarity function is not displayed, it does not look that the minimum of the similarity function can be so confused.

**Table 1.** Above, number of updates based on statistical pattern updating and errors of jumps for different reliability threshold. Below, number of updates in the same sequence using context based pattern updating approach.

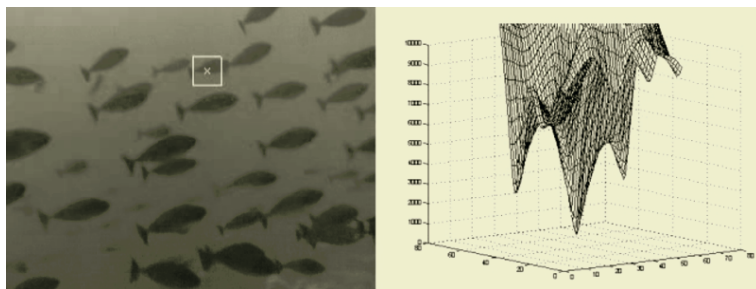
Statistical approach		
Reliability threshold	Number of updates	Jump errors
0.75	-	Yes
0.80	-	Yes
0.85	-	Yes
0.90	468	No

Context based approach		
Reliability threshold	Number of updates	Jump errors
-	106	No

happens due to the existence of local minimums near the absolute one, and all of them surrounded by a very different environment. Such a situation drives to a not pattern updating, and a consequent interest jump error, when the constant threshold and statistic based update algorithms are used.

Things that we perceive or think as quite different may not result be so to a certain similarity function. Figure 2 illustrates such error. Every frame is shown beside its corresponding similarity function. To fix the problem, using the statistic based update method, it is necessary to increase the level of certainty and so the number of updates. Table 1 shows the resulting values of the two compared algorithms. Carrying out both of them a correct tracking process the difference raises in the number of updates needed. The less number of updates the less probability of drifting. The second experiment shows how the proposed updating method can adapt the rate of updates according to the proximity of very similar objects. In figure 3 can be seen a frame of a four seconds sequence where the object of interest is a fish that swims into a shoal, so it is surrounded



**Fig. 3.** The figure shows a frame of a sequence where the object of interest is surrounded by very similar others. The right function depicts the shape of the scene in the representation space. The absolute minimum corresponds to the object of interest and the local minima are objects of the context.

**Table 2.** Number of updates needed by the two methods in order to achieve a correct tracking.

Statistical approach		
Reliability threshold	Number of updates	Jump errors
0.75	-	Yes
0.80	-	Yes
0.85	28	No

Context based approach		
Reliability threshold	Number of updates	Jump errors
-	26	No

by other very similar fishes. In order to avoid the loss of the object, the frequency of updates should be high due to the current pattern can be rapidly confused with objects of the context. The function next to the frame shows graphically the object of interest, as the absolute minimum, and the objects of the context (two fishes) as local minima nearest to the absolute minimum.

## 7 Conclusions

As conclusions from the experiments carried out in a wide range of environments and conditions we can state three major ones:

- The number of required updates is minimized achieving a correct tracking process, and minimizing the drift risk.
- Achievement of an automatic template updating method for any environmental condition.
- The update algorithm is computationally light what allows it to be implemented in low cost general purpose computers.

## References

1. E. H. Adelson and J. R. Bergen. *The plenoptic function and the elements of early vision (Computational Models of Visual Processing, Chapter 1)*. The MIT Press, 1991.
2. M. Andersson. Tracking methods in computer vision. Master's thesis, Computational Vision and Active Perception Laboratory (CVAP), 1994.
3. P. Bellhumeur and D. Kriegman. What is the set of images of an object under all possible illumination conditions? *Int. Journal of Computer Vision*, 3(28):245–260, 1998.
4. Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.
5. V. Colin de Verdière and J. L. Crowley. Visual recognition using local appearance. pages 640–654. *ECCV*, 1998.
6. S. Edelman. Representation is representation of similarities. *Behavioral and Brain Sciences*, 21:449–498, 1998.
7. S. Edelman. *Representation and Recognition in Vision*. The MIT Press, 1999.
8. Hirochika Inoue, Tetsuya Tachikawa, and Masayuki Inaba. Robot vision system with a correlation chip for real-time tracking, optical flow and depth map generation. pages 1621–1629. *IEEE International Conference on Robotics and Automation*, May 1992.
9. Iain Matthews. The template update problem. *IEEE PAMI*, 2004, number =.
10. H. Sebastian Seung and Daniel D. Lee. Cognition: The manifold ways of perception. *Science*, 290:2268–2269, 2000.
11. R. N. Shepard and S. Chipman. Second-order isomorphism of internal representations: Shapes of states. *Cognitive Psychology*, 1:1–17, 1970.

## Part II

# Shape and Matching

# Contour-Based Image Registration Using Mutual Information\*

Nancy A. Álvarez<sup>1</sup>, José M. Sanchez<sup>2</sup>, Jorge Badenas<sup>2</sup>,  
Filiberto Pla<sup>2</sup>, and Gustavo Casañ<sup>2</sup>

<sup>1</sup> Universidad de Oriente, Santiago, Cuba  
aime@fastmail.ca

<sup>2</sup> Universidad Jaume I, Castellón, Spain  
{sanchiz,badenas,pla,ncasan}@uji.es

**Abstract.** Image registration is a problem that arises in many image processing applications whenever information from two or more scenes have to be aligned. In image registration the use of an adequate measure of alignment is a crucial issue. Current techniques are classified in two broad categories: area based and feature based. All methods include some similarity measure. In this paper a new measure that combines mutual information ideas, spatial information and feature characteristics, is proposed. Edge points are used as features, obtained from a Canny edge detector. Feature characteristics like location, edge strength and orientation are taken into account to compute a joint probability distribution of corresponding edge points in two images. Mutual information based on this function is minimized to find the best alignment parameters. The approach has been tested with a collection of portal images taken in real cancer treatment sessions, obtaining encouraging results.

## 1 Introduction

Image registration techniques find applications in several medical fields, like tissue or injury evolution monitoring. In some medical applications there is a need of integrating complementary information from different imaging sensors, that is, different radiological imaging modalities, and also in matching images from the same modality taken at different times.

Portal imaging consists of sensing therapeutic radiation applied from electron accelerators in cancer treatment [1]. They are formed by the projections of anatomical structures over the sensing area after it goes through the body. Due to the high energy of the radiation, there is a poor contrast in portal images compared to x-ray, axial tomography or magnetic resonance images. Introduction of electronic portal imaging devices has increased the quality of portal images.

Detection of patient pose errors during or after treatment is the main use of portal images. For patient pose monitoring, portal images are compared to higher quality simulated portal images used as reference, or to a reference portal image taken at the first therapy session. Any misalignment has to be detected and corrected. Misalign-

---

\* Work partially supported by the Spanish Ministry of Science and Technology under Project TIC2003-06953, and by *Fundació Caixa Castelló* under project P1-1B2002-41.

ments are traditionally detected manually after the session. Automatic misalignment detection before the session, using an innocuous dose, is desirable.

Several registration methods with some degrees of automation, designed to compare portal images among them and with their corresponding simulation images, have been reported in the literature [2, 3, 4, 5].

This work is being developed as part of a project between the Radiotherapeutical Oncology Department at Provincial Hospital of Castellón, Spain, and University Jaume I, Castellón. It is aimed at automating and improving quality control in radiotherapy, mainly focused at patient positioning. We describe a registration method based on ideas of mutual information. Instead of a joint probability distribution derived from grey levels, used in conventional mutual information registration, we propose a joint probability function derived from the spatial localization of features, and features similarity. The minimization of the mutual information based on this function provides the alignment parameters between two images. The method has been tested with portal and magnetic resonance images.

## 2 Related Work

Registration algorithms have applications in many fields. They are valuable tools in medical imaging, remote sensing, computer vision, etc. Currently, research is directed to multimodal registration and to cope with region deformations [6].

Many different registration algorithms have been proposed, and almost all share a common framework: optimizing a cost function that measures the alignment between images [7]. In feature-based approaches the cost function is computed from characteristics of features (edges, ridges) extracted before registration. In the case of portal images, features from the irradiation field geometry have been used [8], where the distance measure is based on the Hausdorff distance modified by using a voting scheme that is expressed as a parameter introduced in the expression of this distance. This modification makes the method tolerant to small position errors like those that occur with automatic edge detectors. Techniques that use manually selected landmarks to be matched have been also used, [9]. In this work contours of the irradiation field are manually selected and their points used for registration using chamfer matching [10].

Pixel-based approaches use all the pixels of an image. A Fourier transform-based cross correlation operator was used in [4] to find the optimal registration, accounting for translations and rotations. A new image alignment measure was introduced in [11, 12] based on entropy concepts developed as part of Shannon's information theory: *mutual information*. It was used to measure the statistical dependence between image intensities of corresponding pixels in two images.

Hybrid techniques that combine both approaches have been proposed. In [13] mutual information is computed using feature points locations instead of image intensity. In [14] the registration function includes spatial information by combining mutual information with image gradient.

Our method uses edges detected from portal images from conventional edge extractors. The registration function is derived from the mutual information concept, and combines three attributes of edges: edge point location, edge strength and edge orien-



tation. These attributes provide spatial information, and are used to build a probability estimate of the possible correspondence of two edge points in two images. A joint probability table is computed for all possible correspondences, and minimization of the mutual information is applied to obtain the best match and the alignment parameters.

### 3 Registration Based on Entropy Minimization

#### Mutual Information

Mutual Information is a concept from information theory, and is the basis of one of the most robust registration methods [15]. The underlying concept of mutual information is *entropy*, which can be considered a measure of dispersion of a probability distribution. In thermology, entropy is a measure of the disorder of a system. A homogeneous image has a low entropy while a high contrast image has a high entropy. If we consider as a system the pairs of aligned pixels in two images, disorder, and joint entropy, increases with misregistration, and correct alignment gives a minimum of the mutual information of the two images.

Given two images  $A$  and  $B$ , the definition of the mutual information  $I(A,B)$  is:

$$I(A,B) = H(A) + H(B) - H(A,B), \quad (1)$$

$H(A)$  and  $H(B)$  being the entropies of images  $A$  and  $B$ , and  $H(A,B)$  being the joint entropy. Correct registration corresponds with maximization of the mutual information. Following Shannon's information theory, the entropy of a probability distribution  $P$  is computed as:

$$H = -\sum_{p \in P} p \log p. \quad (2)$$

Typically, the joint probability distribution of two images is estimated as a normalized joint histogram of the intensity values [12]. The marginal distributions are obtained by summing over the rows or over the columns of the joint histogram.

#### Including Feature Information

We propose a new measure of mutual information computed only from features. We use edge points as features, and point location, edge strength and edge orientation as feature characteristics. Edge points are a significant source of information for image alignment, they are present in portal images and in simulated radiographies obtained from a treatment planner, so they are useful for intra and inter modality registration. In optimal alignment position edge points from one image should match their corresponding points in location and also in edge strength and orientation.

In [13] a new mutual information-based measure was introduced. Instead of using image intensity for estimation of mutual information it uses feature points location information. Let  $\{a_1, a_2, \dots, a_N\}$  and  $\{b_1, b_2, \dots, b_M\}$  be two sets of feature points in two images  $A$  and  $B$ . The mutual information is a function of the joint probability:

$$I(p) = \sum_i \sum_j p_{ij}^T \log \frac{p_{ij}^T}{\sum_k p_{kj}^T \sum_l p_{il}^T}, \quad (3)$$

where  $p_{ij}$  represents the joint probability between feature point  $i$  in  $A$  and feature point  $j$  in  $B$ :

$$p_{ij}^T = \frac{e^{(-\lambda D_{ij}^T)}}{\sum_{ij} e^{(-\lambda D_{ij}^T)}} \tag{4}$$

$T$  stands for the spatial mapping (rigid, similarity, affine) applied for aligning one point set with the other.  $D_{ij}^T$  is a distance measure between two points  $a_i$  and  $b_j$  (e.g. Euclidean distance).  $p_{ij}^T$  is a measure of the correspondence likelihood between those two feature points, while  $\sum_i p_{ij}^T$  and  $\sum_j p_{ij}^T$  are the marginal probabilities.

In [14] the mutual information measure is extended to include spatial information. Locations with valuable spatial information (e.g. transition of tissues) are denoted by strong gradients. The extension is accomplished by multiplying the mutual information extracted from grey level probability distributions with a gradient term. This term includes the gradient magnitude and orientation. The mutual information measure proposed in [14] is:

$$I_{new}(A,B) = G(A,B) I(A,B) \tag{5}$$

with  $G(A,B)$  being the gradient term obtained as:

$$G(A,B) = \sum_{(a_i, b_j) \in A \cap B} \alpha (\nabla a_i, \nabla b_j) \min(|\nabla a_i|, |\nabla b_j|) \tag{6}$$

$a_i$  and  $b_j$  denote two points in images  $A$  and  $B$ , and  $\alpha$  is the angle between two gradient vectors.

When the two images are registered, point  $a_i$  will be located close to its matching point  $b_j$ . If a joint probability table is built considering the distances from each  $a_i$  to all the  $b_j$  with  $j=1, 2, \dots, M$ , in one of the  $M$  cells of the  $i$ -th column, there will be a maximum of that column, point  $b_j$ , so having the biggest likelihood of being the match of  $a_i$ . Re-computing the table for different spatial mappings  $T$ , one of the joint probability tables obtained will be the best, having the smallest distances of matched points. Similarly, with the images registered, an edge point  $a_i$  will match some  $b_j$  having similar edge strength since they represent the same edge point. The edge orientation after the mapping has to be also similar.

Denoting as  $D_{ij}$  the distance between  $a_i$  and  $b_j$ ,  $\Phi_{ij}$  the difference in edge strength, and  $O_{ij}$  the difference in edge orientation after the mapping, we can base the mutual information measure on these feature points characteristics:

$$I(A,B) = f(D_{ij}, \Phi_{ij}, O_{ij}) \tag{7}$$

Our main contribution is the use of several feature attributes to estimate the joint probabilities. We use the gradient magnitude at a feature point as an estimation of the edge strength and the gradient direction as an estimation of the edge orientation:

$$D_{ij}^T = \|a_i - b_j^T\|^2, \quad \Phi_{ij} = \left| |\nabla a_i| - |\nabla b_j| \right|, \quad O_{ij}^T = \cos^{-1} \frac{\nabla a_i \nabla b_j^T}{|\nabla a_i| |\nabla b_j^T|} \tag{8}$$

Gradient magnitude at edge points can be different in corresponding edges detected in different images due to the possibly different sensing devices used to take the images.

This can be overcome by scaling the gradient magnitude at edges in both images, giving, for example, a relative measure between zero and one.

To estimate the joint probability of match between two edge points in two images we introduce an exponential function based on the feature attributes. If  $D_{ij}^r$ ,  $\Phi_{ij}$ ,  $O_{ij}^r$  are small, there is a higher probability of correspondence between those edge points. The proposed joint probability is expressed as follows:

$$p_{ij}^r = \frac{\exp\left(-\left(\frac{D_{ij}^r}{\gamma_1} + \frac{\Phi_{ij}}{\gamma_2} + \frac{O_{ij}^r}{\gamma_3}\right)\right)}{\sum_i \sum_j \exp\left(-\left(\frac{D_{ij}^r}{\gamma_1} + \frac{\Phi_{ij}}{\gamma_2} + \frac{O_{ij}^r}{\gamma_3}\right)\right)} \quad (9)$$

with  $\gamma_k$  being constants. Using the probability distribution function given in (9), mutual information is computed as described in (3).

The main advantage of our approach compared to the classical mutual information is that this latter method does not use the neighbouring relations among pixels at all, all spatial information is lost, while our approach is precisely based on spatial information. Compared to the method reported in [13], we propose a combination of feature attributes, compared to the method in [14], our approach is only based on feature points.

## Edge Detection

Extraction of edges can be done by several methods, first derivative-based methods (Sobel masks), or second derivative-based, like Laplacian of a Gaussian or Canny [16]. In this work we have used the Canny edge detector, that selects edge points at locations where zero-crossings of the second derivative occur.

## Optimization

Optimization of the registration function is done by exhaustive search over the search space. We assume a rigid transformation to align one image with the other, a rotation followed by a translation, both in 2D, so the search space is three-dimensional.

A revision of optimization strategies can be found in [17]: Powell's method, and simplex method, conjugate-gradient and Levenberg-Marquardt methods. Since the principal purpose of our work is to prove the feasibility of a new form of obtaining the joint probability used for the computation of the mutual information, no analysis on the convenience of using a certain optimization has been made.

Exhaustive search is a sufficiently simple method for a bounded three-dimensional search space, and it finds a global optimum, avoiding the main drawback of other optimization algorithms of converging to a local optimum.

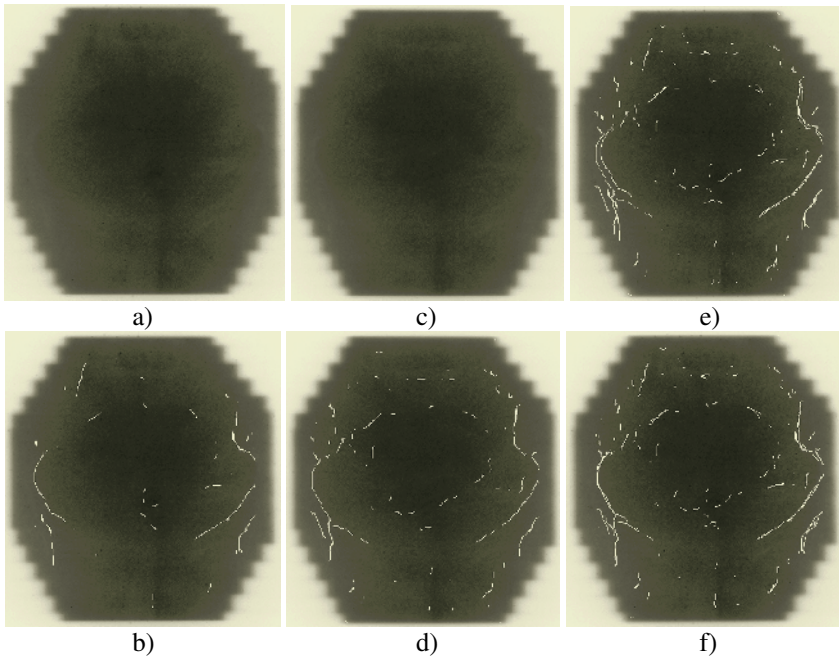
## 4 Results

We have tested our approach with about fifteen pairs of medical images of different sources, portal images provided from sessions of radiotherapy treatments at Provincial Hospital of Castellón, and Magnetic Resonance (MR) images obtained from the internet [18].

For portal image registration, image alignment parameters were determined by human operators and compared to the results of our approach. For MR images the alignment parameters were available along with the images.

The influence of each feature in (9) was tested by making several experiments where each term:  $D_{ij}^T$ ,  $\Phi_{ij}^T$ ,  $O_{ij}^T$ , was included or not. The overall best performance was observed when the three characteristics are used.

In the computation of  $p_{ij}^T$  the values of  $\gamma_1$ ,  $\gamma_2$  and  $\gamma_3$  were fixed heuristically (20, 10, and 1). They were selected by computing  $e^{D_{ij}^T}$ ,  $e^{\Phi_{ij}^T}$  and  $e^{O_{ij}^T}$  using the edge sets without applying any transformation, and observing the graphical representation of these functions. As our intention was that small values of  $D_{ij}^T$ ,  $\Phi_{ij}^T$ ,  $O_{ij}^T$  represent a high correspondence probability, we fixed  $\gamma_1$ ,  $\gamma_2$  and  $\gamma_3$  as values close to the time constants of the exponential functions.

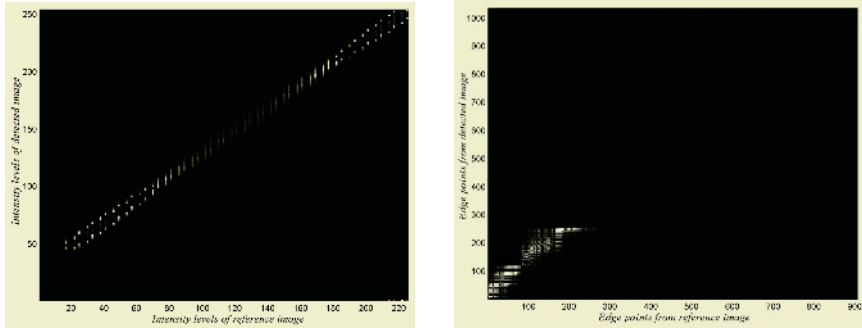


**Fig. 1.** Portal image of a hip with patient in a) initial position and in c) wrong position. Edges detected in each image, b) and d). Both sets of overlapped edges e) before and f) after the registration

Figure 1 shows the registration of two portal images. 2a) shows a portal image of a hip taken in an initial radiotherapy session. c) shows an image from another session that must be aligned with a). b) and d) show the same images with its edges superimposed. These edges correspond mainly to hip bones. e) shows the same image as a) with edges overlapped from both b) and d) before registration, and f) reflects the

arrangement of edges after registration with our approach. Note the improvement in alignment.

Figure 2 compares the joint probability functions obtained after applying the classical mutual information approach using a joint histogram of grey levels, and our feature-based method. Both functions are computed for the best alignment parameters obtained from the registration, that is, lowest entropy. High intensity values correspond to high likelihood of correspondence.



**Fig. 2.** Joint probability functions computed with classical grey level-based mutual information approach (left) and our feature-based method (right)

We have tested our approach with several portal images and compared the results with the registration given manually by several human observers, and with the classical mutual information method based on grey level values. The manual registration was made by identifying common features in both images and registering them. The errors of our method with the human results were within acceptable levels, often less than a pixel in translation and less than a degree in rotation, and always better than the classical method. Although we assumed a rigid transformation in our tests, there is no a priori restriction to a particular type of transformation.

## 5 Conclusions and Further Work

The inclusion of spatial data in the computation of the mutual information is a subject under current investigation. In this paper we propose a new measure of registration that combines mutual information with spatial data obtained from feature attributes, like edge points. Instead of a joint histogram of grey levels, the classical approach, we estimate a joint probability distribution based on feature points. We introduce a probability estimate that two feature points match based on points similarity. An optimization algorithm is then applied to find the best registration parameters where a minimum of the mutual information based on joint entropy occurs.

The proposed approach can be used to register images from different sources, multimodal registration, since it can combine different features as needed. One has to provide a way to compute the probability that two features in two images correspond.

The new approach improves the classical mutual information method, based only on intensity values, which obtains poor performance in low contrast images like portal

images. Furthermore, the number of points used to build the probability function is significantly smaller, only feature points, than the number used to build the joint histogram in the classical approach, all points in the images.

Further work is addressed at investigating the use of other features in the approach, as boundaries of regions in segmented images, or the overlapping area of segmented regions. The key question is which attributes to include in the computation of the joint probability table, and how to combine them.

## References

1. Langmack, A.: Portal Imaging. *Br J Radiol* 74 (2001) 789-804
2. Gottesfeld, L.: A survey of image registration techniques. *ACM Computing Surveys* 24 (1992) 325-376.
3. Plattard, D., Soret, M., Troccaz, J., Vassal, P., Giraud, JY., Champleboux, G., Artignan, X., Bolla, M.: Patient Set-Up using Portal Images: 2D/2D Image Registration Using Mutual Information. *Computer Aided Surgery* (2000) 246-262
4. Hristov, DH., Fallone, BG.: A gray-level image alignment algorithm for registration of portal images and digitally reconstructed radiographs. *Med Phys* 23 (1996) 75-84
5. Van Elsen, PA., Pol, E., Viergever, M.A.: Medical image matching – a review with classification. *ACM Computing Surveys* 24 (1992) 325-376
6. Lester, H., Arrige, S.R.: A survey of hierarchical non-linear medical image registration. *Pattern Recognition* 32 (1999) 129-149
7. Zitová, B., Flusser, J.: Image registration methods: a survey. *Image and Vision Computing* 21 (2003) 977-1000
8. Chmielewski, L., Kukulowicz, P.F., Gut, P., Dabrowski, A.: Assessment of the quality of radiotherapy with the use of portal and simulation images – the method and the software. *Journal of Medical Informatics & Technologies* 3 (2002) 171-179
9. Leszczynski, K., Loose, S., Dunscombe, P.: Segmented chamfer matching for the registration of field borders in radiotherapy images. *Phys Med. Biol* 40 (1995) 83-94
10. Borgfors, G.: Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Trans PAMI* 10 (1988) 849-865
11. Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., Suetens, P.: Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging* 16 (1997) 187-198
12. Viola, P., Wells, W.M.: Alignment by maximization of mutual information. *International Journal of Computer Vision* 24 (1997) 137-154
13. Rangarajan, A., Chui, H., Duncan, J.: Rigid point feature registration using mutual information. *Medical Image Analysis* 4 (1999) 1-17
14. Pluim J., Maintz, J.B., Viergever, M.A.: Image registration by maximization of combined mutual information and gradient information. *IEEE Transactions on Medical Imaging* 19 (2000) 809-814
15. West, J. et al.: Comparison and evaluation of retrospective intermodality brain image registration techniques. *J. Comput. Assited Tomography* 21(1997) 554-566.
16. Canny, J.F: A Computational Approach to Edge Detection. *IEEE TPAMI* 8 (1986) 679-698
17. Maes, F., Vandermeulen, D., Suetens, P.: Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information. *Medical Image Analysis* 3 (1999) 373-386
18. <http://www.itk.org>

# Improving Correspondence Matching Using Label Consistency Constraints

Hongfang Wang and Edwin R. Hancock

Dept. of Computer Science, University of York  
Heslington, York, YO10 5DD, UK  
{hongfang, erh}@cs.york.ac.uk

**Abstract.** In this paper we demonstrate how to embed label consistency constraints into point correspondence matching. We make two contributions. First, we show how the point proximity matrix can be incorporated into the support function for probabilistic relaxation. Second we show how the label probabilities delivered by relaxation labelling can be used to gate the kernel matrix for articulated point pattern matching. The method is evaluated on synthetic and real-world data, where the label compatibility process is demonstrated to improve the correspondence process.

## 1 Introduction

Finding one-to-one feature point correspondences is a challenging problem in the matching of deforming shapes. Many existing approaches rely on the method. For example, point distribution models (PDMs) [1] require reliable one-to-one feature point correspondences over a sequence of examples for the purposes of learning the modes of shape variation. The factorisation method of Tomasi and Kanade [10] also requires accurate correspondence information to separate motion and shape. Without an accurate means of locating the feature correspondences, the recovered model will be inaccurate. For instance, in point distribution models then the covariance matrix will represent the distribution of correspondence errors rather the modes of shape variation. In the case of factorisation, then there will be errors in both the estimated motion and the recovered shape. In the literature, many attempts have been described to recover accurate correspondences. For example, in [3] the softassign method is used to compute correspondences in a manner that is robust to outliers.

Of course there is a wealth of information that can be exploited to improve and refine the point-correspondence process. If absolute position is used, then the detailed transformation between point sets must be recovered. This is of course straightforward if the point sets are known to undergo a rigid transformation, for example affine or perspective, or if there is a well defined non-rigid transformation that can be applied, for example a spline-warp or a diffeomorphism. One way of avoiding the need to know the transformation geometry is to characterise the point-sets using information concerning their relational arrangement. For instance, here proximity graphs [12] or proximity matrices [8, 9] can be used. The points can also be augmented using neighbourhood feature characteristics [5].

An additional but little used source of information is that provided by label consistency constraints. In many types of image, the points can be assigned semantic labels to distinguish their identity. The simplest of these distinguishes whether the point is foreground or background. A more complex example would be to assign labels to distinguish the object subparts, for example the limbs of articulated objects. Using this information the consistency of pairwise relations can be tested against a scene constraint model. Hence, correspondences which are inconsistent with the model can be rejected.

In this paper we aim to develop a method which allows label consistency constraints to be incorporated into the point correspondence process. To do this we draw on ideas from probabilistic relaxation labelling [2, 4, 6]. We characterise each point by augmenting the positional information with a vector of label probabilities. In addition, the arrangement of the points is represented using a Gaussian point proximity matrix. Our first contribution is to show how the point-proximity matrix can be incorporated into the definition of the support function for relaxation labelling. In this way when the label probabilities are updated, then the strength of the proximity relations is brought to bear on the computation of label support. Our second contribution is to show how the label probabilities can be used to refine the point correspondence process. Here we draw on our previous work [11] and use a kernelised version of the Shapiro and Brady algorithm [9]. We use the label probabilities to refine the kernel matrix used to locate point correspondences. The matching process is realised as an iterative process which has interleaved steps for label probability update and point correspondence matching.

## 2 Label Process

One of the best studied approaches in the literature for labeling problems in computer vision involves using relaxation techniques [2, 4, 6]. Relaxation labelling can either be an “offline” belief propagation process that distributes the previously learned labeling confidence to the whole feature set, e.g., [4], or an “online” learning process that learns the labeling information on the fly, e.g., [2, 6]. In a discrete relaxation processes, initially each node is assigned all possible labels and inconsistent labels are discarded in the process until a consistent label distribution is obtained. In the continuous or probabilistic case, each node is assigned an initial weight or probability distribution. Iteratively, the label probabilities or weights are updated, again until a consistent distribution is reached. However, whichever labeling process is used, the performance depends critically on the compatibility coefficients and the support function used to combine evidence in the iterative process. In [2], a dictionary is used, and in [4] the compatibility coefficients are represented as a vector which is learned offline. Here our compatibility model shares some factors in common with the compatibility vector in [4].

The labeling process that we develop here is an evidence combining one that propagates label constraints. Our label consistency information is totally



contained within a “model” feature point-set. We will hence learn the label compatibility information from the model point-set, before attempting to match it against the data. Our first step is to collect label information from the model image, and apply the learned label compatibility model in the second step of the process which involves assigning consistent point labels to the “data” point-set.

We are interested in matching two point-sets  $\mathbf{y} = \{\mathbf{y}_i\}_{i=1}^n$ ,  $\mathbf{y}_i = (y_{i1}, y_{i2})$ , and  $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^m$ ,  $\mathbf{x}_i = (x_{i1}, x_{i2})$ . To apply semantic constraints to the feature points, we augment the feature point vectors with a label probability vector for the independent rigid component present in the scene. We specify the point-set  $\mathbf{y}$  as the model point-set and its label probability values are given beforehand. Accordingly, the point-set  $\mathbf{x}$  are called the “data” point-set where the label probabilities are to be assigned. Assume there are  $L$  labels in each image. An image point  $\mathbf{x}_i$  can be assigned to a label  $\theta_i \in \Omega$ , where  $\Omega = \{\omega_i\}_{i=1}^L$ . Denote by  $P(\theta_i = \lambda)$  the probability that node  $\mathbf{x}_i$  is labeled as  $\lambda$  with  $\lambda \in \Omega$ . Then the vector  $\mathbf{p}_i = (P(\theta_i = \omega_1), \dots, P(\theta_i = \omega_L))^T$  represents the probabilities of assigning each of the possible labels to the point  $\mathbf{x}_i$ , with  $0 \leq P(\theta_i = \lambda) \leq 1$ , and  $\sum_{\lambda=1}^L P(\theta_i = \lambda) = 1$ . The matrix  $P$  with the probability vectors as columns, i.e.,  $P = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N)^T$ , represents the label probability distribution over the entire point-set. The individual point-sets are characterised using a Gaussian proximity matrix  $W$ . For the point-pair  $\mathbf{x}_i$  and  $\mathbf{x}_j$  the element of the matrix is given by:

$$W_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2) \quad (1)$$

where  $\|\mathbf{x}_i - \mathbf{x}_j\|^2$  is the Euclidean distance between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , and  $\sigma$  is a constant width parameter of the Gaussian function.

## 2.1 Label Compatibility Information

Our aim is to develop a relational description of the point-sets using information concerning point proximity and a label compatibility matrix. The compatibility matrix  $R \in \mathbb{R}^{L \times L}$  is of dimension  $L \times L$  and embodies knowledge of the number of rigid components, i.e. labels, in each image, and the semantic constraints between each pair of object-labels. The matrix has elements  $R_{ij} = 1$  if  $\mathbf{x}_i$  and  $\mathbf{x}_j$  come from the same rigid part, and is  $-1$  otherwise. This definition restricts the nodes to give total positive support to the nodes in the same group and contribute a negative support to nodes outside the group. The proximity constraint is also acquired from the model image. We assume that in any two consecutive image frames, the relative position of the rigid components of the object under study will not change significantly.

## 2.2 Label Update Formula

The label probabilities for the data point-set are updated iteratively commencing from a set of initial values. With the label compatibility information learned from the model point-set, we update the label probability for each point according to the support from its neighbourhood. Let us denote the neighbourhood

for the point  $\mathbf{x}_i$  and its  $k$  closest points by  $N_i = \{\mathbf{x}_{i1}, \dots, \mathbf{x}_{ik}\}$ . Here we use the Euclidean distance between the points  $\mathbf{x}_i$  and  $\mathbf{x}_j$  to define the neighbourhood. With these ingredients the support from the neighbourhood for the label assignment  $\lambda_i$  to point  $\mathbf{x}_i$  is:

$$S_{i,\lambda_i} = \frac{\exp\{\sum_{j \in N_i} \sum_{\lambda_j \in \Omega} P(\theta_j = \lambda_j) R(\lambda_i, \lambda_j) W_{ij}\}}{\sum_{\lambda_i \in \Omega} \exp\{\sum_{k \in N_i} \sum_{\lambda_k \in \Omega} P(\theta_k = \lambda_k) R(\lambda_i, \lambda_k) W_{ik}\}} \quad (2)$$

where  $R(\lambda_i, \lambda_j)$  are the entries of the label compatibility matrix  $R$  which measure the compatibility of the label pairs  $\lambda_i$  and  $\lambda_j$ . Here the elements of the proximity matrix  $W$  are defined as in (1) and are used to weight the label-support.

Having defined the support equation, the label probabilities are updated using the formula:

$$P^{(n+1)}(\theta_i = \lambda) = \frac{P^{(n)}(\theta_i = \lambda) + \beta S_{i,\lambda}^{(n)}}{\sum_{\lambda_i \in \Omega} (P^{(n)}(\theta_i = \lambda) + \beta S_{i,\lambda}^{(n)})} \quad (3)$$

where  $\beta$  is a constant parameter.

### 3 Matching

Our aim is to combine the label information with the proximity information and to develop a point association matching process to locate the feature point correspondences. Our idea is to use a kernel function to map the data into a possibly higher dimensional feature space, and then perform an eigen-decomposition on the covariance matrix of the data in this space to construct the data mapping. The matching process is further gated by the label-probabilities obtained in the previous algorithm step. Assuming that the data  $\mathbf{x}$  are already centered, this covariance matrix is given by  $\overline{C} = \frac{1}{m-1} \sum_{i=1}^m \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T$ . The eigensystem is recovered by solving the equation  $m\lambda\alpha = K\lambda$ , where  $K$  is the Gram matrix with its entries obtained from a kernel function  $k(\mathbf{x}_i, \mathbf{x}_j)$ . In this paper the Gaussian kernel function is used in the experiments. The  $p_{th}$  feature vector, corresponding to the projection of the  $p_{th}$  feature point into the eigenspace, takes the form

$$\langle v^p, \Phi(\mathbf{x}) \rangle = \sum_{i=1}^m \alpha_i^p k(\mathbf{x}_i, \mathbf{x}) \quad (4)$$

To generalize the method to non-centered data, the covariance matrix  $\overline{C}$  needs to be re-computed. In our case where more than one rigid component is present in the data point-set, the data need to be centered onto their respective subpart centre of movement. Thus, the mean value of each data group in the feature space needs to be computed and subtracted from  $\Phi(\mathbf{x}_i)$ . For the group with label  $\lambda$ , the mean-position (i.e. subgroup centre) is given by

$$\mu_\lambda = \frac{1}{\sum_i P(\theta_i = \lambda)} \sum_i \Phi(\mathbf{x}_i) P(\theta_i = \lambda). \quad \text{for each } \lambda \in \Omega$$

Denote by  $\tilde{\Phi}(\mathbf{x}_i) = (\Phi(\mathbf{x}_i) - \sum_{\lambda} \mu_{\lambda} P(\theta_i = \lambda))$ , the covariance matrix of the centered data is given by  $\tilde{C} = \frac{1}{m-1} \sum_{i=1}^m \tilde{\Phi}(\mathbf{x}_i) \cdot \tilde{\Phi}(\mathbf{x}_i)^T$ , where

$$\begin{aligned} \tilde{\Phi}(\mathbf{x}_i) \cdot \tilde{\Phi}(\mathbf{x}_i)^T &= k(\mathbf{x}_i, \mathbf{x}_i) \\ &- \sum_{\lambda \in \Omega} \frac{P(\theta_i = \lambda)}{\sum_j P(\theta_j = \lambda)} \sum_j P(\theta_j = \lambda) k(\mathbf{x}_i, \mathbf{x}_j) \\ &- \sum_{\lambda \in \Omega} \frac{P(\theta_i = \lambda)}{\sum_k P(\theta_k = \lambda)} \sum_k P(\theta_k = \lambda) k(\mathbf{x}_k, \mathbf{x}_i) \\ &+ \sum_{\lambda \in \Omega} \frac{P^2(\theta_i = \lambda)}{\sum_j P(\theta_j = \lambda) \sum_k P(\theta_k = \lambda)} \sum_j \sum_k P(\theta_j = \lambda) P(\theta_k = \lambda) k(\mathbf{x}_k, \mathbf{x}_j) \end{aligned} \quad (5)$$

After computing the feature vectors  $\tilde{\mathbf{y}}_i$  and  $\tilde{\mathbf{x}}_j$  using (4) for the respective model and data point-sets, the next step is to compute the association  $M_{ij} = \exp(-d_{ij}^2/\sigma)$  of the point pairs. Denote the label agreement of the point pair  $\mathbf{y}_i$  and  $\mathbf{x}_j$  by  $P(\theta_j = \lambda, \theta_i = \lambda, \forall \lambda \in \Omega)$ , the association of the two feature vectors is further gated by this constraint:

$$\tilde{M}_{ij} = P(\theta_j = \lambda, \theta_i = \lambda, \forall \lambda \in \Omega) M_{ij}, \quad (6)$$

The correspondences are defined as the most similar node pairs. That is, for each node  $\mathbf{x}_i$  in the data point-set, the correspondence in the model set is the node  $\mathbf{y}_j$  that has the largest association  $M_{ij}$  with  $\mathbf{x}_i$ . Assume that the labels on each feature point are independent of each other, then the consistency of the label on point  $\mathbf{x}_i$  and the label on  $\mathbf{x}_j$  is computed by:

$$P(\theta_i = \lambda, \theta_j = \lambda, \forall \lambda \in \Omega) = \sum_{\lambda=1}^L P(\theta_i = \lambda) P(\theta_j = \lambda) \quad (7)$$

The matching process is thus an iterative one in which at each step new label probabilities are incorporated to improve the matching. Since as an increasing number of correspondences are found, the value of  $S = \sum_i \exp(\|\mathbf{x}_i - \mathbf{y}_i\|_F/2\sigma^2)$ , where we assume  $\mathbf{x}_i$  and  $\mathbf{y}_i$  are the correspondence pair from data point-set and model point-set, respectively, will increase, and ultimately reach a maximum value. The matching process contains the following steps:

1. Initialize  $P, t, S_{old}$ ;
2. Compute the Gaussian association matrix for each point-set;
3. Run the labeling process, compute  $P^{new}$ ;
4. Use  $P^{new}$  to compute  $\tilde{C}_{new}$  using equation (5) and (6);
5. Compute  $M$  and find the current correspondences;
6.  $\text{diff} = S - S_{old}$  or  $\text{iteration} < t$ ;
7. if  $\text{diff} < \text{threshold}$ 
  - return;
- else
  - update  $P$  using the matching results.
- end
8. Go to step 2.

## 4 Experimental Results

We experiment with both synthetic data and real world data sets. The purpose of using the synthetic data-sets is to test the algorithm under conditions of controlled noise. Three groups of synthetic data are generated. Two data-set pairs contains three rigid components, and the third pair two components. The point-sets are shown in Figure (1). The experiment is designed to simulate articulated object movement with the rigid components undergoing individual motion. To do this, in each pair, the second data set is obtained from an articulated motion, where the data in the second and third components are transformed by the equation  $X' = sRX + \mathbf{t}$ , where  $s$  is a scaling parameter,  $\mathbf{t} = (t_x, t_y)'$  is the translation vector, and  $R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$  is the rotation matrix. In the second component,  $s = 0.8$ ,  $\mathbf{t} = (10, 15)'$ ,  $\theta = 20^\circ$ , and in the third component,  $s = 1.2$ ,  $\mathbf{t} = (10, 15)'$ ,  $\theta = 30^\circ$ . In both experiments, the initial label probability distributions are chosen to be uniform. Figure (1) and Table (1) show the result of the labeling process.

**Table 1.** Matching and labeling results (error%)

Data-set	Num of points	Num of labels	Gaussian matching	Articulated matching(1)*	Articulated matching(2)**	Labeling
1	100	3	98	29	25	11
2	60	2	88.3	30	30	0
3	30	2	36.7	13.3	13.3	3.3
4	55	2	38.2	25.2	25.2	0
5	51	3	86.27	35.3	29.41	2
Note:	*: Results based label information from the label process					
	**: Results obtained when correct label information is assumed.					

We simulate the effects of data uncertainty in the following way. First, Gaussian noise  $G \sim \mathcal{N}(0, \Sigma)$  is added to each second data-set, so as to simulate random position jitter. Second, points are deleted in data-set 5. From the results we can see that this algorithm accommodates these uncertainties well. Figure (2) shows the effect of noise with varying covariance matrices, and a different percentage of points missing. Figure (3) also shows the matching result.

## 5 Discussion

The labeling process described above can be improved to be more robust and flexible. In the framework of relaxation labeling, the compatibility coefficients play an important role in making the labeling precise. The current compatibility matrix will be improved in future work to accommodate semantic constraints more effectively.

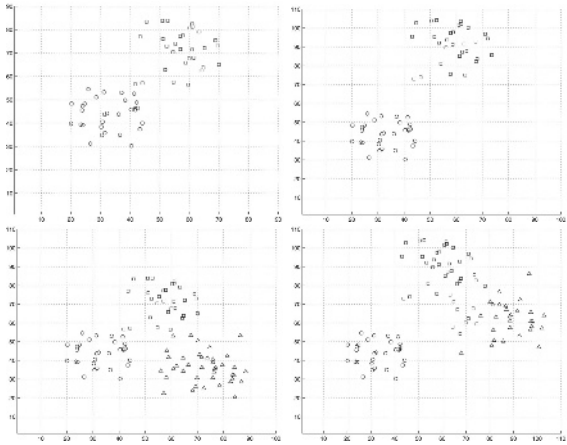


Fig. 1. Synthetic data, labeling result. Top: data-set pair 2; Bottom: data-set pair 1

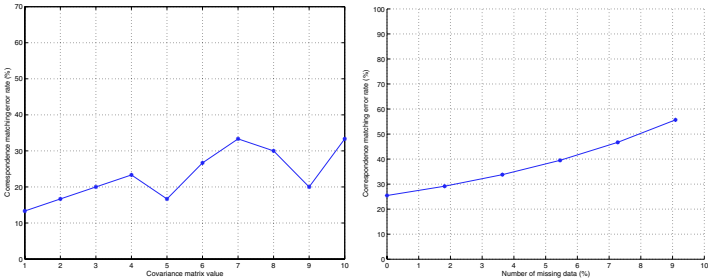


Fig. 2. Left: The effect of adding Gaussian noise with varying  $\Sigma$  (data-set 3). Right: The effect of missing points in the second point-set (data-set 5)

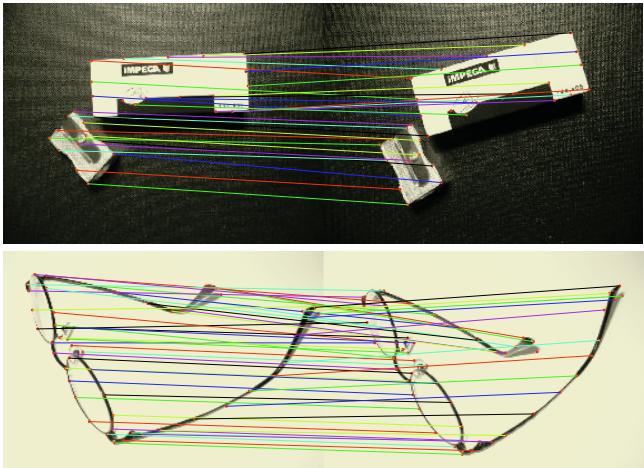


Fig. 3. Matching result: data-set 3 (top) and 5 (bottom)

It is well known that the Gaussian kernel is invariant under similarity transformations since it is based on pairwise Euclidean distances. Given that the relative positions of each rigid component are not affected significantly, the kernel matching process we developed in prior work can be used to recover a reasonably accurate set of one-to-one point correspondences and provide good initial label probability estimates. This can also be considered as a route to improving the efficiency of the labelling process. Also other kernels which are capable of extracting the transformation invariants and are insensitive to structural errors may be used in the matching process to improve the results.

## Acknowledgements

This work was supported by the UK MOD Corporate Research Programme. The authors thank Dr. Andrew R. Webb for his encouragement and support.

## References

1. T. F. Cootes and C. J. Taylor and D. H. Cooper and J. Graham. Training Models of Shape from Sets of Examples. *In Proceedings BMVC 1992*, pp.9-18.
2. J. Kittler and E. R. Hancock. Combining Evidence In Probabilistic Relaxation. *International Journal of Pattern Recognition And Artificial Intelligence*, Vol.3, No.1, pp.29 - 51, 1989.
3. S. Pappu, S. Gold and A. Rangarajan. A framework for non-rigid matching and correspondence. *Advances in Neural Information Processing Systems*, vol.8, pp.795 - 801, 1996.
4. M. Pelillo and M. Refice. Learning Compatibility Coefficients for Relaxation Labeling Processes. *IEEE Trans.PAMI*, Vol.16, No.9, pp.933 - 945, 1994.
5. M. Pilu. A direct method for stereo correspondence based on singular value decomposition. *IEEE CVPR 1997*.
6. A. Rosenfeld and R. Hummel and S. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Systems. Man and Cybernetics*, Vol.6, pp.420 - 433, 1976.
7. B. Schölkopf, A. J. Smola and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, vol.10, pp.1299-1319, 1998.
8. G. L. Scott and H. C. Longuet-Higgins. An Algorithm for Associating the Features of Two Images. *Proc. Royal Soc. London Series B*, vol.244, pp.21 - 26, 1991.
9. L. S. Shapiro and J. M. Brady. Feature-Based Correspondence - An Eigenvector Approach. *Image and Vision Computing*, vol.10, pp.283-288, 1992
10. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137-154, 1992.
11. H. Wang and E. R. Hancock. A kernel view of spectral point pattern matching. *In proceedings S+SSPR*, LNCS vol.3138, pp.361-368, 2004.
12. R. C. Wilson and E. R. Hancock. Structural Matching by Discrete Relaxation. *IEEE Trans. PAMI*, vol.19, No.6, pp.634-648, 1997.

# The Euclidean Distance Transform Applied to the FCC and BCC Grids

Robin Strand

Centre for Image Analysis, Uppsala University  
Lägerhyddsvägen 3, SE-75237 Uppsala, Sweden  
robin@cb.uu.se

**Abstract.** The discrete Euclidean distance transform is applied to grids with non-cubic voxels, the face-centered cubic (fcc) and body-centered cubic (bcc) grids. These grids are three-dimensional generalizations of the hexagonal grid. Raster scanning and contour processing techniques are applied using different neighbourhoods. When computing the Euclidean distance transform, some voxel configurations produce errors. The maximum errors for the two different grids and neighbourhood sizes are analyzed and compared with the cubic grid.

## 1 Introduction

Three-dimensional images are usually captured in the cubic grid. The main reason is tradition and that the data structure is easy to handle. It is, however, possible to adjust image capturing techniques such as CT or MRI to produce images in other grids, such as the face-centered cubic (fcc) and body-centered cubic (bcc) grids. It has been demonstrated that the hexagonal grid in two dimensions is theoretically better than the square grid, [1]. The fcc and bcc grids are the three-dimensional “equivalents” of the hexagonal grid, [2]. When applying a Distance Transform (DT), each object (background) grid point is assigned the distance to the closest background (object) grid point. DTs are very important in the field of image analysis with many applications such as, e.g., skeletonization, watershed segmentation, and template matching. In this paper, the properties of the Euclidean DT on the fcc and bcc grids are examined.

The most common way to compute a DT is to use a raster scanning (Chamfering) algorithm, [3–5]. With this technique, the image is scanned sequentially two or more times. In each scan, the distance values are propagated through the image. To compute a *weighted* DT in a cubic grid, two scans are sufficient, and to compute the *Euclidean* DT, four scans are necessary, [6]. Raster scanning algorithms to compute the weighted DT for the fcc and bcc grids were examined in [7]. The raster scanning technique to compute the Euclidean DT is applied to the fcc and bcc grids in Section 3.

Another way to compute the Euclidean (or weighted) DT is to use contour processing, or ordered propagation. The basic idea is to iteratively propagate distance values by starting with the grid points on the contour of the object and

considering neighbours of already visited grid points. This is done by, in each iteration, constructing a list of all grid points that are to be included in the next iteration. This technique has been examined in, e.g., [8–10]. Its generalization to the fcc and bcc grids is examined in Section 4.

There are basically two other techniques to compute the Euclidean DT. A parallel algorithm introduced by Yamada, [4, 11], and Voronoi diagram construction algorithms, [12–14]. These techniques are not considered in this paper. In the parallel algorithm, all grid points are processed in each iteration which makes it inefficient on standard computers. Since the algorithm is inherently parallel, it performs well only on parallel computers. The Voronoi diagram construction algorithms are too complex to be of any practical interest for these grids.

The DTs produced by the algorithms examined in this paper are not totally error-free; a thorough error analysis is performed in Section 5. Many different approaches to extend these algorithms in order to make the DTs error-free has been made. One way of doing this is to employ extra processing in regions where errors are likely to appear, see e.g. [10], where the neighbourhoods are enlarged or [9], where extra iterations are performed. One way to make the contour processing algorithm error-free is to keep the propagation chain convex by splitting it when it loses its convexity, see [8].

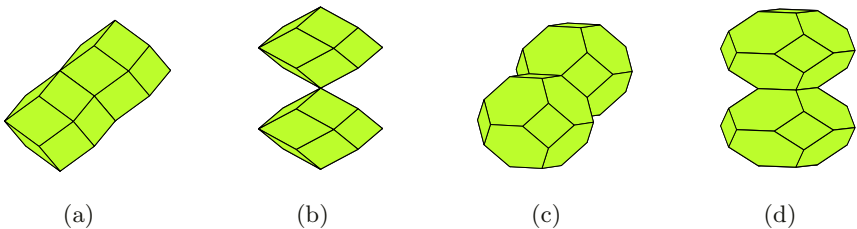
## 2 The Grids

The grids that are examined are the fcc,  $\mathbb{F}$ , and the bcc,  $\mathbb{B}$ , grids defined as

$$\mathbb{F} = \{(c_1, c_2, c_3) : c_1, c_2, c_3 \in \mathbb{Z} \text{ and } c_1 + c_2 + c_3 \equiv 0 \pmod{2}\} \text{ and}$$

$$\mathbb{B} = \{(c_1, c_2, c_3) : c_1, c_2, c_3 \in \mathbb{Z} \text{ and } c_1 \equiv c_2 \equiv c_3 \pmod{2}\}.$$

The adjacencies used in the fcc grid are the 12-adjacency (face-neighbours) and the 18-adjacency (face- and vertex-neighbours), see Fig. 1. In the bcc grid, the neighbours connected to a grid point are all face-neighbours, see Fig. 1. However, there are two kinds of face-neighbours, which results in the 8-adjacency and the 14-adjacency.



**Fig. 1.** Voxels corresponding to: (a) 12-adjacent grid points in fcc, (b) 18-adjacent grid points in fcc, (c) 8-adjacent grid points in bcc, and (d) 14-adjacent grid points in bcc



### 3 Sequential Algorithm

In this section, the raster scanning technique is discussed. The basic idea is to scan the image from one corner of the image to the opposite corner. First, background grid points are set to 0 and object grid points in the distance image are set to  $\infty$  for the weighted DT and a vector  $(\infty, \infty, \infty)$  for the Euclidean DT. For each grid point  $p$ , the minimum distance value of all weights/vectors at grid points in the *mask* ( $q_i$ ) plus the weight/vector associated with  $q_i$  relative to  $p$  is computed, see, e.g., [3–6]. In the following,  $x+ y+ z+$  denotes the loop

```

for( $x = 1 : x_{\max}$ )
  for( $y = 1 : y_{\max}$ )
    for( $z = 1 : z_{\max}$ )
      for( $\mathbf{v} \in \text{mask}_{x+y+z+}$ )
        if ( $\|\mathcal{I}((x, y, z) + \mathbf{v}) - \mathbf{v}\|_2 < \|\mathcal{I}(x, y, z)\|_2$ )
           $\mathcal{I}(x, y, z) = \mathcal{I}((x, y, z) + \mathbf{v}) - \mathbf{v}$ 
        end
      end
    end
  end
end,

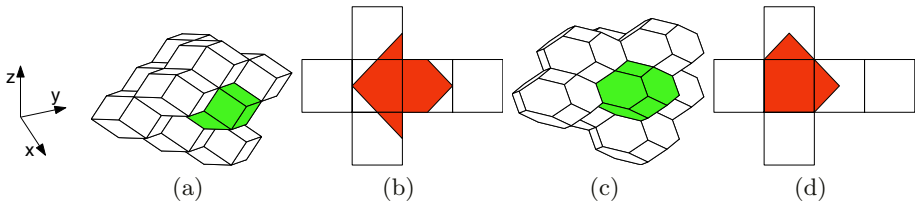
```

where  $\mathcal{I}$  denotes the distance image. This loop is a part of the sequential algorithm used to compute the Euclidean DT.

When computing a weighted DT, two scans are sufficient for the cubic grid. For the fcc and bcc grids, some caution is needed, see [7]. Since, in a raster scan, two consecutive grid points are not adjacent when using the 12-adjacency in fcc or the 8-adjacency in bcc, these adjacencies are not suited for a DT based on raster scanning. Therefore, only the 18-adjacency in fcc and the 14-adjacency in bcc are considered in this section.

When computing the Euclidean DT, two scans are not sufficient, [6]. This is because the relative coordinates between grid points are propagated through the image and, contrary to the weighted DTs where a limited number of weighted prime directions are considered, the shortest path between the grid points can have any direction. To examine the number of scans needed and what masks that should be used in each scan, the the Unfolded Cube Graph (UCG) is used. The UCG was introduced by Ragnemalm in [6]. Using the UCG, it is possible to analyze which neighbours that are needed for the propagation in each image scan. Given a mask, the corresponding UCG is constructed by unfolding a cube in  $\mathbb{R}^3$  with the central grid point centered in the cube. The directions that are supported by the mask are mapped on the cube. In two dimensions, this is done by considering the unit circle. Since it is hard to visualize the complete surface of a sphere, a cube is used instead.

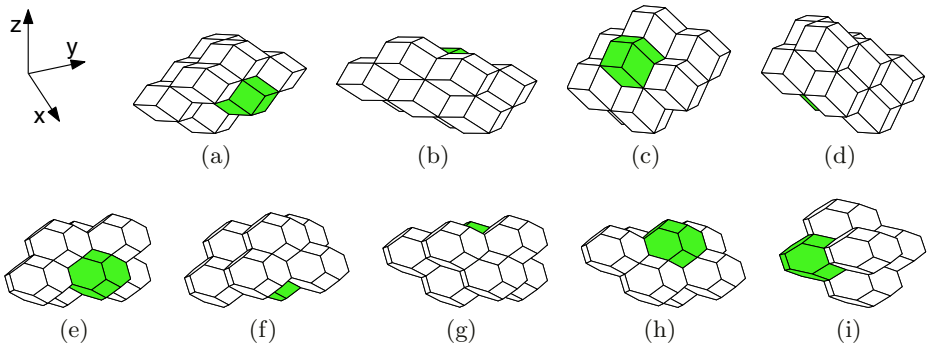
In a raster scan, only grid points that have already been visited are useful in the mask. For both the fcc and the bcc grids, the maximal mask that can be used in each scan is exactly half the set of neighbouring grid points, see Fig. 2. The amount of the UCG they fill up are also shown in Fig. 2.



**Fig. 2.** The largest masks that can be used in the propagation  $x+ y+ z+$  in fcc (a) and bcc (c) and the corresponding Unfolded Cube Graphs (b) and (d), respectively

The mask in fcc covers  $7/24$  of the UCG, so the least number of scans are four. In bcc, the mask covers  $3/12$  of the UCG. This suggests that exactly four scans are needed. This is, however, not the case. The reason is the geometry of the region covered in the UCG. It is not possible to cover the cube with four regions shaped as in Fig. 2(d) when using masks consisting of grid points that are positioned so that they can propagate distance information.

The UCG is used to examine in which order the image should be scanned. If, when folded, the areas that correspond to a set of masks fill the whole surface of the cube, then these masks and the corresponding scanning directions constitutes sufficient raster scans. The whole direction space is filled up if the scanning is performed as  $x+ y+ z+$ ,  $x- y+ z-$ ,  $z+ y- x-$ ,  $z- y- x+$  (fcc) and  $x+ y+ z+$ ,  $z- y+ x+$ ,  $x- y+ z-$ ,  $z+ y+ x-$ ,  $y- x+ z+$  (bcc) with the masks as in Fig. 3.



**Fig. 3.** The masks in fcc (a-d) and bcc (e-i) derived using UCG

## 4 Contour Processing

When using a contour processing algorithm to compute the Euclidean DT, the vector pointing to the nearest background grid point is propagated. In the first step, a dynamic list of all contour grid points, i.e. the object grid points with an adjacent grid point in the background, is constructed. All grid points in the list

are processed by propagating the vector to the closest background grid point to its adjacent neighbours. The neighbours are now put in a new list and in the next iteration, only the grid points in the new list are considered. The algorithm stops when a vector pointing to the closest background grid point has been assigned to all object grid points. See also [8–10].

The contour processing technique is easy to generalize to the fcc and bcc grids. It works for all adjacencies and the only errors that appears are those examined in Section 5.

## 5 Error Analysis

In this section, the grid is considered as a subset of  $\mathbb{R}^3$ . Because of the discrete structure of the grid, the configurations that produces errors are limited. Therefore, the theoretical maximum error in  $\mathbb{R}^3$ , as calculated in e.g. [4], is of limited practical interest. Instead, the configuration in the grid that actually produces the maximum error is calculated. To be sure that the maximum error is found, it is, however, necessary to do some calculations in  $\mathbb{R}^3$ .

Given an adjacency in a grid, the *prime vectors* are the vectors between the center grid point and its adjacent grid points. The error analysis is performed on regions spanned by three adjacent prime vectors, denoted  $p_1, p_2, p_3$ . By performing the error analysis on every region spanned by three adjacent prime vectors, all directions are considered. Thus, the maximum error is obtained.

An error appears in grid point  $\mathbf{0}$  when any of its adjacent grid points does not contain the address to the closest background grid point from  $\mathbf{0}$ . To find the configurations of grid points that give rise to errors, the conditions derived by Mullikin in [15] are used. Given a grid point  $q$  in the region spanned by  $p_1, p_2, p_3$ , an error appears at the origin  $\mathbf{0}$  if there are grid points  $q_1, q_2, q_3$  such that

$$\begin{aligned} \|q_i - p_i\|_2 &< \|q - p_i\|_2 \quad \text{for } i \in \{1, 2, 3\} \\ \|q\|_2 &< \|q_i\|_2 \quad \text{for } i \in \{1, 2, 3\} \end{aligned} \tag{1}$$

This can be reformulated; let  $q$  be a grid point and  $p_1, p_2, p_3$  be three adjacent prime vectors. If there is a grid point in the interior of each of the regions (in  $\mathbb{R}^3$ )

$$\mathcal{R}_i = B(p_i, r_i) \setminus B(\mathbf{0}, s) \quad \text{for } i \in \{1, 2, 3\}, \tag{2}$$

where  $r_i = \|q - p_i\|_2$  and  $s = \|q\|_2$ , then an error is produced by this configuration of grid points. These conditions are shown in Fig. 4 with  $p_1 = (1, 0, 1)$ ,  $p_2 = (0, 1, 1)$ ,  $p_3 = (1, 1, 0)$ , and  $q = (1, 1, 2)$ .  $B(\mathbf{0}, s)$ ,  $B(p_1, r_1)$ ,  $B(p_2, r_2)$ , and  $B(p_3, r_3)$  are denoted  $B1$ ,  $B2$ ,  $B3$ , and  $B4$ , respectively.

Using these conditions, the maximum relative error for a given  $q$  in a grid  $\mathbb{G}$  in the region spanned by  $p_1, p_2$  and  $p_3$  is

$$E(q) = \min_{i \in \{1, 2, 3\}} \left( \max_{q' \in \mathbb{G} \cap \mathcal{R}_i} \left( \frac{\|q'\|_2}{\|q\|_2} \right) - 1 \right). \tag{3}$$

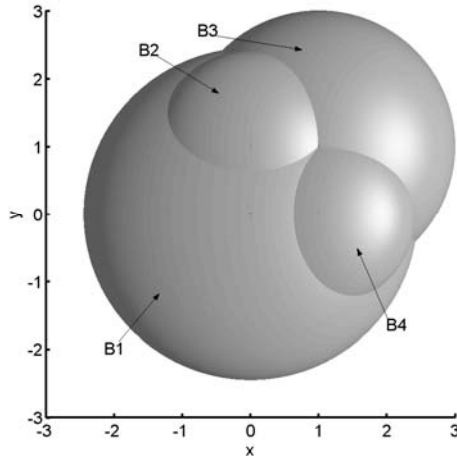


Fig. 4. The conditions in equation (2), see text

To get an absolute upper bound of the relative error, the expression for the error is also given for a  $q$  in  $\mathbb{R}^3$ :

$$\begin{aligned}
 E_{cont}(q) &= \min_{i \in \{1,2,3\}} \left( \sup_{q' \in \mathcal{R}_i} \left( \frac{\|q'\|_2}{\|q\|_2} \right) - 1 \right) \\
 &= \min_{i \in \{1,2,3\}} \left( \frac{\|p_i\|_2 + \|q - p_i\|_2}{\|q\|_2} - 1 \right). \tag{4}
 \end{aligned}$$

The last equality is derived as follows

$$\sup_{q' \in \mathcal{R}_i} (\|q'\|_2) = \sup_{q' \in \mathcal{R}_i} (\|p_i\|_2 + \|q' - p_i\|_2) = \|p_i\|_2 + \|q - p_i\|_2,$$

i.e., the maximum distance from  $\mathbf{0}$  to a point in  $\mathcal{R}_i$  equals the distance to the center of  $B(p_i, r_i)$  plus the radius  $r_i$ . Observe that the error in a grid is equal to zero for most  $q$ . This is not true in the continuous case. In  $\mathbb{R}^3$ ,  $E_{cont}(q) > 0$  and  $E_{cont}(q) \rightarrow 0$  as  $\|q\|_2 \rightarrow \infty$ . With  $q$  at any distance from the origin, (4) is used to derive the following conditions for  $q$  when the error is maximized

$$\|p_1\|_2 + \|q - p_1\|_2 = \|p_2\|_2 + \|q - p_2\|_2 = \|p_3\|_2 + \|q - p_3\|_2. \tag{5}$$

The maximum error is thus found along a curve  $q = l(t), t \in \mathbb{R}_+$ .

An error in a region defined by a set of adjacent prime vectors is obtained by finding the  $q$  closest to the origin that gives an error. This is done by considering only grid points within a given radius from the origin. If there are grid points in each of the regions  $\mathcal{R}_i$  defined in (2), then we get a configuration that gives an error. The error  $E(q)$  is calculated using (3). First, a region defined by a small radius is examined, and then the procedure is repeated for a slightly larger radius until a configuration of grid points that produces an error is found.

Then, the value of  $t = t_0$  such that the error at  $l(t_0)$  in  $\mathbb{R}^3$ ,  $E_{cont}(l(t_0))$ , is equal to  $E(q)$  is calculated. Since the maximum error in  $\mathbb{R}^3$  is found along  $l(t)$ , it is enough to consider the grid points at a distance to the origin smaller than  $\|l(t_0)\|_2$ . This is a cause of the fact that the grid is a subset of  $\mathbb{R}^3$ . Since the maximum error in  $\mathbb{R}^3$  at a distance greater than  $\|l(t_0)\|_2$  never can be larger than  $E(q)$ , this is also true in the grid.

This procedure is explained by a simple example. Consider  $\mathbb{Z}^3$ , using the 6-adjacency. By symmetry it is sufficient to consider  $p_1 = (1, 0, 0)$ ,  $p_2 = (0, 1, 0)$ , and  $p_3 = (0, 0, 1)$ . We compute that with  $q = (1, 1, 1)$ ,  $q_1 = (2, 0, 0)$ ,  $q_2 = (0, 2, 0)$ , and  $q_3 = (0, 0, 2)$  are the grid points closest to the origin that fulfills (1) and where  $\mathcal{R}_i, i \in \{1, 2, 3\}$  in (2) all contain  $q$ . Now, (3) gives  $E(q) = 2/\sqrt{3} - 1 \approx 0.1547$ . By solving (5) for these  $p_1$ ,  $p_2$ , and  $p_3$ ,  $l(t) = (t, t, t)/\sqrt{3}$ . Using (4), we get that with  $t_0 = 2\sqrt{3}$ ,  $E_{cont}(l(t_0)) = 2/\sqrt{3} - 1$ . Thus, the maximum relative error is found within a radius of  $2\sqrt{3}$  from the origin. For these grid points, the maximum error is calculated and the maximum among these is the maximum error that can appear in the grid with this adjacency.

The maximum relative errors for fcc and bcc are listed in Table 1. For comparison,  $\mathbb{Z}^3$  is also included in the table. Note that the errors only appear for some special configurations of grid points satisfying the conditions derived in this section and that in most applications, the errors are neglectible.

**Table 1.** Maximum relative error for EDTs

Grid	maximum error	$q$	$q_1$	$q_2$	$q_3$
$\mathbb{Z}^3$ , 6 nb	15.47%	(1, 1, 1)	(2, 0, 0)	(0, 2, 0)	(0, 0, 2)
$\mathbb{Z}^3$ , 18 nb	4.08%	(2, 2, 2)	(2, 3, 0)	(2, 0, 3)	(0, 2, 3)
$\mathbb{Z}^3$ , 26 nb	2.06%	(4, 2, 2)	(5, 0, 0)	(3, 4, 0)	(3, 0, 4)
$\mathbb{F}$ , 12 nb	6.07%	(4, 0, 0)	(3, 0, 3)	(3, 3, 0)	(3, -3, 0)
$\mathbb{F}$ , 18 nb	4.08%	(4, 4, 4)	(4, 0, 6)	(0, 4, 6)	(4, 6, 0)
$\mathbb{B}$ , 8 nb	9.29%	(6, 0, 0)	(3, 3, 5)	(3, -5, 3)	(3, 3, -5)
$\mathbb{B}$ , 14 nb	2.06%	(6, 0, 6)	(5, 5, 5)	(5, -5, 5)	(7, -5, -1)

## 6 Conclusion

The Euclidean DT has been applied to the fcc and bcc grids. Both raster scanning and contour processing have been considered. To compute the Euclidean DT on the bcc grid, five scans are needed. On the cubic grid and the fcc grid, four scans are sufficient. The result from the error analysis, Table 1, shows that the maximum relative error is highly dependent on the grid and the adjacency that is being used. Very good results are obtained for the bcc grid with 14 neighbours. The reason that the same error is obtained for the fcc grid with 18 neighbours and the cubic grid with 18 neighbours is that the prime vectors are collinear. The error configurations, however, are not completely equivalent. This is because the grid points in the error configuration in the cubic grid are not all grid points in the fcc grid. These results show that the fcc and the bcc grids are well suited for Euclidean DTs.

## Acknowledgement

Many thanks to prof. Gunilla Borgefors for valuable comments during the preparation of this paper.

## References

1. Bell, S.B.M., Holroyd, F.C., Mason, D.C.: A digital geometry for hexagonal pixels. *Image and Vision Computing* **7** (1989) 194–204
2. Herman, G.T.: *Geometry of Digital Spaces*. Birkhäuser, Boston (1998)
3. Borgefors, G.: On digital distance transforms in three dimensions. *Computer Vision and Image Understanding* **64** (1996) 368–376
4. Danielsson, P.E.: Euclidean distance mapping. *Computer Graphics and Image Processing* **14** (1980) 227–248
5. Ragnemalm, I.: The Euclidean distance transform and its implementation on SIMD architectures. In: *Proceedings of 6<sup>th</sup> Scandinavian Conference on Image Analysis*, Oulu, Finland. (1989) 379–384
6. Ragnemalm, I.: The Euclidean distance transform in arbitrary dimensions. *Pattern Recognition Letters* **14** (1993) 883–888
7. Strand, R., Borgefors, G.: Distance transforms for three-dimensional grids with non-cubic voxels. Submitted for publication (2004)
8. Vincent, L.: Exact Euclidean distance function by chain propagations. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii. (1991) 520–525
9. Ragnemalm, I.: Neighborhoods for distance transformations using ordered propagation. *Computer Vision, Graphics, and Image Processing* **56** (1992) 399–409
10. Cuisenaire, O., Macq, B.: Fast Euclidean distance transformation by propagation using multiple neighborhoods. *Computer Vision and Image Understanding* **76** (1999) 163–172
11. Yamada, H.: Complete Euclidean distance transformation by parallel operation. In: *Proceedings 7<sup>th</sup> international Conference on Pattern Recognition*, Montreal. (1984) 69–71
12. Maurer, C.R., Qi, R., Raghavan, V.: A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25** (2003) 265–270
13. Breu, H., Gil, J., Kirkpatrick, D., Werman, M.: Linear time Euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17** (1995) 529–533
14. Guan, W., Ma, S.: A list-processing approach to compute Voronoi diagrams and the Euclidean distance transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (1998) 757–761
15. Mullikin, J.C.: The vector distance transform in two and three dimensions. *CVGIP: Graphical Models and Image Processing* **54** (1992) 526–535

# Matching Deformable Regions Using Local Histograms of Differential Invariants\*

Nicolás Pérez de la Blanca<sup>1</sup>, José M. Fuertes<sup>2</sup>, and Manuel J. Lucena<sup>2</sup>

<sup>1</sup> Department of Computer Science and Artificial Intelligence  
ETSII. University of Granada, 18071 Granada, Spain  
nicolas@ugr.es

<sup>2</sup> Departamento de Informática. Escuela Politécnica Superior. Universidad de Jaén  
Avenida de Madrid 35, 23071 Jaén, Spain  
{jmf, mlucena}@ujaen.es

**Abstract.** This paper presents a technique to enable deformable regions to be matched using image databases based on the information provided by the differential invariants of local histograms for the key-region. We shall show how this technique is robust enough to deal with local deformations, viewpoint changes, lighting changes, large motions of the tracked object and small changes in image rotation and scale. The proposed algorithm is based on the building of a specific template where an orthogonal representation space is associated with each of its locations. This space is calculated from neighboring information provided by a vector of local invariants calculated on each of the image's pixels. Unlike other well-known color-based techniques, this algorithm only uses the pixels' gray level values.

## 1 Introduction

In this paper, we shall explore the problem of matching deformable image regions using image databases or image sequences. The basic information used in our work is provided by local histograms of a finite set of image-bands defined from invariant values calculated on the image. What is new about our approach is the template definition which provides us with a very robust approach for dealing with local shape and lighting deformations. Deformable object matching remains a very challenging problem mainly due to the absence of good templates and similarity measures which are robust enough to handle all the geometrical and lighting deformations that might be present in a matching process.

The use of invariant features to match or index objects from images is a well-known approach in computer vision although originally, this was mainly used to characterize objects from their outline shape [11]. In order to recognize objects from their pixel values, different geometrical and lighting differential invariants have been suggested [5],[16],[18]. In practice, however, this type of invariant has only proved useful when applied on points with rich geometrical structures in their neighborhoods [9],[12]. In [6] and [7], a new type of image is introduced where each pixel has an

---

\* This work is partially supported by Grant TIC2001-3316 from the Spanish Ministry of Science and Technology.

associated histogram of values rather than a scalar value. This new image concept is the inspiration for our approach, and we shall use it to associate to each pixel a summary of the information defining its neighborhood. In our approach, local histograms obtained after applying each invariant on all image pixels are used as the local features characterizing the neighboring region of each pixel. Our approach is region-based since spatial features better model the type of application we are interested in. Let us consider facial region matching. In recent years, object recognition by parts has been suggested as a very efficient approach for recognizing deformable objects [1],[3],[4]. Although different approaches are used in the recognition process from basic features, the selection and detection of good features is a common task shared by all approaches.

The use of histograms as features of interest can be traced back to Swain & Ballard [17] who demonstrated that color histograms could be used as a robust and efficient mechanism for indexing images in databases. Histograms have been used widely in object and texture recognition and image and video retrieval in visual databases [2],[3],[14]. The main drawback of using global histograms as the main feature is the loss of spatial information. Recent approaches based on the space-scale theory have incorporated the image's spatial information. In [14], multidimensional histograms obtained by applying Gaussian derivative filters to the image are used. This approach incorporates the image's spatial information with global histograms. In [2], while spatial information is also taken into account, a set of intensity histograms are used at multiple resolutions. In [8], it is shown how extremely relevant information for detecting salient regions in the image can be extracted from local histograms at different scales. None of these approaches, however, explicitly addresses the use of the local spatial invariant information present in the image.

In this paper, unlike the approaches mentioned above, we shall attempt to achieve a better compromise between spatial information and robustness to deformations. In our case, the matching template for each image region is built as a spatial array, and a set of histograms (calculated from a spatial neighborhood centered on this position) is associated to each of its positions. Each of these histograms defines a new axis of the representation space associated to each pixel. Building a new orthogonal representation of this space and extracting only the most relevant axis a new parsimonious orthogonal representation of it can be obtained. The projection of the histograms into the new orthogonal subspace provides the coefficient vector used in the matching process. On each image, the template is iterated on all the possible locations within it. The matching score on each image location is the Euclidean norm of the vector difference between the projection coefficients associated to the image and the template, respectively.

This paper is organized into five sections: Section 2 presents the template definition and the matching process; Section 3 presents the gray value invariants we have used in the experiments; Section 4 shows the experimental results; and finally, Section 5 details the discussion and conclusions.

## 2 Template Definition and the Matching Process

Let  $\mathcal{R}$  be a region of an image  $I$ , defining our region of interest (ROI). Different gray level invariants can be calculated on each  $\mathcal{R}$  location according to the geometrical or



lighting transformation groups that we expect to deform the region. Standard template matching techniques based on these invariants, however, need only be shown to be effective if applied on image location with a rich gray level structure such as that given by a corner [9][16]. The technique we introduce to characterize the ROI uses a different approach.

Let  $ni$  be the number of different independent invariants to be calculated on each  $\mathcal{R}$  pixel location. Let  $\mathbf{IB}(\mathcal{R}) = \{\mathbf{IB}_1, \mathbf{IB}_2, \dots, \mathbf{IB}_{ni}\}$  be the set of band-images calculated by applying each invariant to the region  $\mathcal{R}$ . An  $nbin \times ni$  matrix,  $\mathbf{H}_1$ , is associated to each pixel location of our ROI where the columns of this matrix are the local histogram in a neighborhood of the pixel from each of the  $\mathbf{IB}$  matrices. The bin number,  $nbin$ , is fixed beforehand and all the histograms are normalized to this value. Each histogram is calculated from a fixed size neighborhood around the pixel.

The set of histograms associated to a pixel can be considered as the different axes of a space characterizing the pixel neighborhood information. According to the gray level structure around the pixel, some of the invariant values provide more relevant information than others. In order to obtain an orthogonal parsimonious representation of this space, we calculate the singular value decomposition on the  $\mathbf{H}_1$  matrix,  $\mathbf{H}_1 = \mathbf{U}\mathbf{D}\mathbf{V}^T$  and we select the  $s$  columns  $\mathbf{U}_s = \{\mathbf{U}_1, \dots, \mathbf{U}_s\}$  associated to the  $s$  highest singular values as the new axis of the space. A threshold on the normalized singular values ratio is used to select the most significant ones. The projection of the  $\mathbf{H}_1$  matrix into this new space  $\mathbf{U}_s$  is given by:

$$\mathbf{c}(x,y) = \mathbf{U}_s^T(x,y) \cdot \mathbf{H}_1(x,y) \tag{1}$$

The  $\mathbf{c}(x,y)$  matrix provides us with the set of coefficients characterizing the pixel location  $(x,y)$ . In the matching process, we start by calculating  $\mathbf{H}_1$  on each pixel location  $(r,s)$  of the target image. We then calculate a similarity measure on each  $(r,s)$  location by shifting the image template on the target image. The similarity measure is given by:

$$\mathbf{S}(r,s) = \left\{ \sum_{x,y} \left\| \mathbf{c}(x,y) - \mathbf{U}_s^T(x,y) \mathbf{H}'_t(x+r,y+s) \right\| \right\} \tag{2}$$

where the sum is on all pixel locations  $(x,y)$  of the region-template. The matrices  $\mathbf{c}(x,y)$  and  $\mathbf{U}_s^T(x,y)$  correspond to the template location  $(x,y)$  and the matrix  $\mathbf{H}'_t$  to the target image in location  $(x+r,y+s)$ . The estimated target location is given by the location of the minimum value of  $\mathbf{S}$  and we use the Euclidean norm.

In our case, all the local histograms are very sparse since the range of gray levels present in the neighborhood of each pixel is usually very small in comparison with the full range of the image. One important consequence of this situation is the need to quantize the image's gray level range before the similarity distances are calculated. A consequence of the quantization process is the invariance to illumination changes which are smaller than the bin width. In all of our experiments, we use a uniform quantization criterion fixing the same length to the interval of the gray levels assigned to each bin. The same process is applied to the gray levels of the template region.

### 3 Gray Level Invariants

In this paper, we use the set of invariants suggested by Schmid in [16]. We only use differential invariants based on the three first order derivatives of the image. The following table shows the invariants used in our experimentation in tensor notation:

$$\begin{aligned}
 v_S(1, \dots, 8) &= \begin{bmatrix} L_i L_i \\ L_i L_j L_j \\ L_{ii} \\ L_{ij} L_{ji} \\ \varepsilon_{ij} (L_{jkl} L_i L_k L_l - L_{jkl} L_i L_j L_l) \\ L_{ij} L_j L_k L_k - L_{ijk} L_i L_j L_k \\ -\varepsilon_{ij} L_{jkl} L_i L_k L_l \\ L_{jkl} L_i L_j L_k \end{bmatrix}, \quad v_L(1, \dots, 3) = \begin{bmatrix} L_i L_j L_j \\ (L_i L_i)^{3/2} \\ L_{ii} \\ (L_i L_i)^{1/2} \\ L_{ij} L_{ji} \\ L_i L_i \end{bmatrix}, \quad v_L(4, \dots, 7) = \frac{1}{(L_i L_i)^2} \begin{bmatrix} v_S(5) \\ v_S(6) \\ v_S(7) \\ v_S(8) \end{bmatrix} \quad (3) \\
 v_L(8) &= \frac{\varepsilon_{ij} \varepsilon_{kl} L_i L_j L_k L_l}{(L_m L_m)^{3/2}}, \quad v_L(9) = \frac{\varepsilon_{ij} L_j L_k L_{kl}}{(L_m L_m)^{3/2}}
 \end{aligned}$$

where  $v_S$  represents the differential invariants associated to the SO(2) similarity group,  $v_L(1:7)$  represents the associated invariant to gray level affine transformations, and  $v_L(8:9)$  represents two invariants associated to lighting reversible transformation [5]. The Cartesian expression of the invariants can be obtained using the usual conventions:

$$\begin{aligned}
 L_x &= \frac{\partial L}{\partial x}, \quad L_i = \sum_i L_i = L_x + L_y, \quad L_{ij} = \sum_{i,j} L_{ij} = L_{xx} + L_{xy} + L_{yx} + L_{yy} \\
 \varepsilon_{11} &= \varepsilon_{22} = 0, \quad \varepsilon_{12} = -\varepsilon_{21} = 1
 \end{aligned} \quad (4)$$

### 4 The Algorithm

The previous steps can be summarized as follows:

- 1.- Fix the scale value for the histograms.
- 2.- Fix the set of invariants to be used and calculate their associated image-bands.
- 3.- Calculate the local-histogram matrix on each location of the template region.
- 4.- Build up the template  $\mathcal{T}(\mathcal{R}_g)$  of the region template using SVD on each of the local-histogram matrices.
- 5.- For each target image:
  - 5.1 Build the local-histogram matrix on each location of the image.
  - 5.2 Shift the template frame on all possible image locations. On each location to project the local-histogram matrices on the orthogonal spaces of the corresponding template location to calculate the image coefficients  $\mathbf{c}(\mathbf{x}, \mathbf{y})$ .
  - 5.3 Calculate the similarity measure associated to each template position using (2).
  - 5.4 Take the image location with the  $\mathbf{S}(\mathbf{x}, \mathbf{y})$  minimum value as the best target location.

In order to increase the efficiency of the algorithm, it is applied to a sub-sampled version of the region template and images. From this, we estimate a set of possible points instead of a single location. All these points and their neighboring points (for a fixed size) define the set of points on which we shall apply the algorithm on the original images. The most costly step in this algorithm is the calculation of the similarity maps on each image location. In this respect and taking into account the redundant information present in the template, the error measure given in (2) can only be calculated on a subset of the pixel.

## 5 Experimental Results

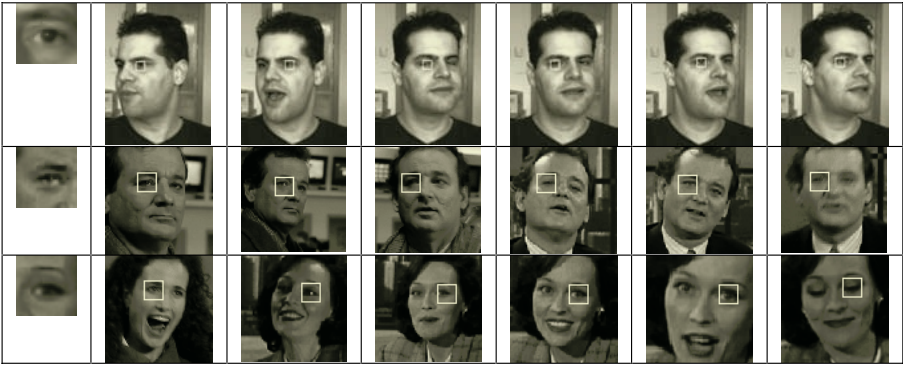
Multiple experiments have been performed in order to assess the effectiveness of the proposed algorithm. Firstly, we have focused our experiments on showing how robust our algorithm is to drastic changes in object pose. Secondly, we have also shown how the algorithm is capable of a reasonable level of shape generalization, since with only one sample it is possible to successfully match different instances of the same kind of object. Thirdly, we have shown how robust our algorithm is when there is a very large change in pose and a very hard noise condition. In all of the experiments, we have used a frame with a seven-pixel radius for the histogram estimation. We also quantify the entire histogram range to 32 bins. The active range of the invariant images is selected using a saturation threshold on the invariant values. In our case, a range of values between 100 and  $-100$  was used. In all the experiments, the template region is a rectangular sub-image. In all the experiments, we have tried with different sampling steps (0-4) on the image axis in order to calculate the expression in (2). In all the images, a sampling step of 4 pixels in both axes was sufficient to obtain the highest saliency value in the best location. The full set of the 17 differential invariants has been used in all the experiments.

Video sequences of human heads in motion and two sequences obtained from the Oxford face database<sup>1</sup> have been used in our experiments. Our recorded sequences have one-hundredth images. The head in motion sequences were captured in 640x480 format by the same digital camera, but in different lighting conditions. For reasons of efficiency in our experiments, we reduce the image size to the head zone giving 176x224 size images. The Oxford Groundhog-Day database comprises 243 images, which we split into two different sets with men's and women's faces, respectively. The pictures from the Oxford database are 81x81 pixels. Our aim is to match the eyes and the mouth throughout the entire sequence. In our case, the template region was an instance of the matched object chosen from an image of the sequence. However, we also show the results of using a fixed template region on a different image sequence.

In the different rows in Figure 1, we show relevant results for three different sequences where the goal is to match the eye region. The image template for each row is shown in the first cell of the row. The first row shows a person moving their head from right to left as they change their facial expression. The second and third rows show results from the Oxford face database. Figure 2 shows relevant results from the mouth matching experiments. As in Figure 1, the first row shows an image from a

---

<sup>1</sup> <http://www.robots.ox.ac.uk/~vgg/data4.html>



**Fig. 1.** In this figure, the results of the eye-matching problem are shown. In each row, the region-template used is shown in the first column. The white rectangle indicates the best matching region.



**Fig. 2.** This figure shows relevant results for the mouth-matching problem. In each row, the region-template used is shown in the first column. The first row shows images from a recorded sequence and the second and third row show the results from the Oxford database. The white rectangle indicates the best matching region.

recorded sequence and the second and third row show the results from the Oxford database.

The experiments show how our algorithm is stable and robust enough for view-point changes, local deformations, moderate scale changes and illumination changes. The images in both figures show how our template is flexible enough to match very different instances of an object. This means that the template definition is capable of codifying the relevant information about the object by removing local spatial details. It is also important to emphasize that the algorithm in our experiment is over 90% efficient when the template region and the images are from the same person, but when we match a region template from one person with images from another person, efficiency drops to between 50%-60%. This indicates a lack of generalization that could be explained by the set of used invariants. It is also relevant to point out that the presented results have been obtained when the template-regions cover not only the particular feature of interest but also part of its surrounding area.

In all the experiments, we have only considered translation motions of the template since we are interested in showing that the proposed algorithm is capable of successfully matching a large set of different instances of the original template. Of course, the inclusion of motions such as rotation or scale should greatly improve the technique. One of the main drawbacks of our algorithm is the loss of the image-plane rotation invariance that is present when the full image histogram is considered.

## 6 Conclusions

In conclusion, we have proposed a new matching algorithm for the case of deformable regions and shown its application to face region matching. This algorithm enables us to match different instances of the same object by making use of the information provided by a set of geometrical and lighting invariants. The loss of local order imposed by the use of local histograms has resulted in a high level of robustness in template matching with strong shape deformations even in high noise conditions and moderate lighting changes. Although in theory the algorithm is not robust enough for image-plane rotation and scale, experiments have also shown that there is invariance to small rotations and scale. Full invariance to scale could be obtained by applying a space-scale approach. This, together with achieving higher invariance to lighting changes, shall be one of our future lines of research.

## References

1. S. Agarwal and D. Roth, Learning a sparse representation for object detection, ECCV'02, 113-130, 2002
2. E.Hadjidemetriou, M.D. Grossberg and S.K. Nayar: Spatial information in multiresolution histograms, In Intern. Conf. CVPR'01, 2001.
3. B.Heisele, P.Ho, J.Wu and T.Poggio, Face recognition: component-based versus global approaches, Computer Vision and Image Understanding 91,6-21,2003
4. R.Fergus, P. Perona and A. Zisserman: Object class recognition by unsupervised scale-invariant learning. In IEEE CVPR'03, 264-271, 2003.
5. L.M.J. Florack, B.M. ter Haar Romeny, J.J. Koenderink and M.A. Viergever, Scale and the differential structure of images, Image and Vision Computing, vol-10, 6, 1992.
6. L.D.Griffin, Scale-imprecision space, Image and Vision Computing 15, 369-398, 1999.
7. J.J.Koenderink and A.J. Van Doorn: The Structure of locally orderless images, Intern. Journal of Computer Vision 31 (273), 159-168, 1999.
8. T. Kadir and M. Brady: Scale, saliency and image description. Intern. Journal of Computer Vision, 45 (2):83-105, 2001.
9. D.G.Lowe, Object recognition from local scale-invariant features. In ICCV'99, 1150-1157.
10. J.Matas, O.Chum, M.Urban and T.Pajdla: Robust wide baseline stereo from maximally stable extremal regions. In BMCV'02 Conference, 384-393, 2002.
11. J.L. Mundy and A. Zisserman (eds), Geometric invariance in computer vision, MIT Press, 1992.
12. K.Mikolajczyk and C. Schmid: An affine invariant interest point detector. In ECCV'02, 128-142, 2002
13. W. Niblack. The QBIC project: Querying images by content using color, texture and shape. In Proc. Of SPIE Conf. on Storage and Retrieval for image and video database, vol-1908, 173-187, 1993.

14. B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In ECCV'96, Vol I, pages 610--619, 1996.
15. B. Schiele and J. L. Crowley: Robustness of object recognition to view point changes using multidimensional receptive fields histograms. ECIS-VAP, 1996.
16. C. Schmid and R. Mohr, Local greyvalue invariants for image retrieval, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 19-5, 530-535, 1997.,
17. M.J. Swain and D.H. Ballard. Color Indexing Intern. Journal of Computer Vision, 7(1):11-32.1991.
18. B.M. ter Haar Romeny, L.M.J. Florack, A.H. Saldem and M.A. Viergever, Higher order differential structure of images, Image and Vision Computing, vol-12, 6, 1994.
19. T.Tuytelaars and L. Van Gool: Wide baseline stereo based on local affinity invariant regions, In British Machine Vision Conference, Bristol, U.K.,412-422. 2000

# A Global-to-Local Matching Strategy for Registering Retinal Fundus Images

Xinge You<sup>1,2</sup>, Bin Fang<sup>1</sup>, Zhenyu He<sup>1</sup>, and Yuan Yan Tang<sup>1</sup>

<sup>1</sup> Department of Computer Science

Hong Kong Baptist University

{xyou, fangb, zyhe, yytang}@comp.hkbu.edu.hk

<sup>2</sup> Faculty of Mathematics and Computer Science, Hubei University, 430062, China  
xyou@hubu.edu.cn

**Abstract.** In this paper, a multi-resolution rigid-model-based global matching algorithm is employed to register tree structures of blood vessels extracted from retinal fundus images. To further improve alignment of the vessels, a local structure-deformed elastic matching algorithm is proposed to eliminate the existence of ‘ghost vessels’ for accurate registration. The matching methods are tested on 268 pairs of retinal fundus images. Experiment results show that our global-to-local registration strategy is able to achieve an average centreline mapping errors of 1.85 pixels with average execution time of 207 seconds. The registration results have also been visually validated by corresponding fusion maps.

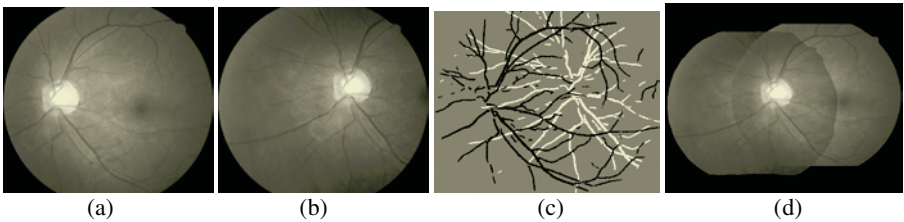
## 1 Introduction

The temporal registration of retinal fundus images is an important application in ophthalmology because a patient is often screened at regular intervals for the development of eye diseases [1]. By comparing the photographs taken at different time periods, physicians can evaluate the progression of the diseases and decide on the appropriate treatments to be taken. Figures 1(a) and 1(b) show the retinal photographs of a patient that have been taken half year apart. With appropriate temporal registration, we are able to highlight the similarities/ differences in the two images (Figure 1(c) and 1(d)).

Alignment methods using the full image content and mutual information as the similarity measure fail to deal with the registration of the total surface of eye fundus images [2]. Bifurcation and intersection points of the vascular tree are usually identified and used as control points for global points mapping based registration methods [3]. However, the detection may be not accurate and the control point number needed to compute a correct transformation may not be sufficient. Therefore, it is expected to use tree structure of blood vessels as object features for retinal fundus registration [4,5]. Vascular trees are typically incorporated into well-established transformation models such as rigid [4], affine [5] and second order polynomial transformations [3]. The process of registration is equivalent to solving the problem of optimizing a function that measures the goodness of fit between the reference and the transformed images. Various search techniques are utilized to find the optimal transformation with

respect to the defined models. [4] used the Powell's method and [5] employed simulated annealing and genetic algorithms.

The sphere-like shape of human eye suggests a quadratic surface model for registration. Nevertheless, the computation complexity of model parameters may be a disaster and the search strategy design is not trivial. Therefore, in this work, we develop the idea to employ a 'global-to-local' matching strategy. First, tree structures of blood vessels are globally aligned using rigid model of translations and rotation. The adoption of relatively simple model enables us to compute the optimal transformation effectively and promptly by multi-resolution global matching technique. On the other hand, local alignment errors occur due to the imprecise model. In order to rectify the pitfall of local misalignment which results in 'ghost vessels', we propose a structure-deformed elastic matching algorithm to improve registration accuracy. 268 pairs of retinal fundus images are used in the experiment to test the proposed methods. The registration algorithms are validated quantitatively by accuracy and efficiency analysis as well as visually by corresponding fusion maps inspection.



**Fig. 1.** Illustration of successful registration for a pair of fundus images with 54.4% overlap. (a) Original gray level image of left eye, (b) Original gray image of the same eye half year later, (c) Overlapped vascular structures before matching, (d) Constructed fusion map.

## 2 Registration Using Vascular Structure

The success of feature-based registrations is largely dependent on the quality of the identified features. For retinal image registration based on extracted vessels, a robust method is needed to detect the vascular structure of retina as reliably as possible. Here, we employ the technique described in [6] which has been demonstrated to be effective in identifying the whole tree structure of blood vessels in retinal images.

### 2.1 Global Multi-resolution Matching

The anatomy of the human eye with its sphere-like shape and physical properties of optical imaging system naturally leads to a quadratic surface model for stereo registration. This is especially important when the images have little overlap due to the large variation in viewpoints between images [3].

However, some important observations in the retinal fundus photograph imaging system suggest the possibility of other suitable transformation models. The use of central retinal images with the same viewpoint reduces perspective effects and indicates that a weak affine model may be sufficient without losing too much accuracy. This motivates us to adopt the weak affine transformation model of translations and



rotation for globally matching two vascular structures of retinal fundus images. The model can be mathematically expressed as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{1}$$

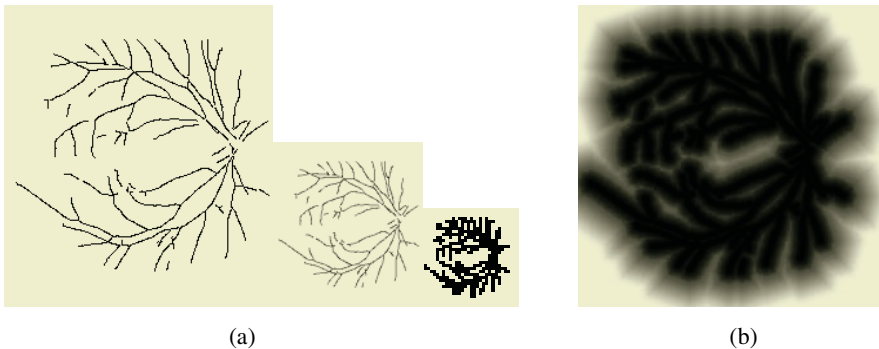
In order to evaluate the goodness of fit between two vascular features, a distance measure is computed in terms of the corresponding transformation. A search for the optimal transformation is to find the global minimum of the defined distance function. The search process typically starts with a number of initial positions in the parameter space.

The idea behind multi-resolution matching is to search for the local optimal transformation at a coarse resolution with a large number of initial positions. Only a few promising local optimal positions with acceptable centreline mapping errors of the resulting transformation are selected as initial positions before proceeding to the next level of finer resolution. The assumption is that at least one of them is a good approximation to the global optimal matching. The algorithm is detailed as follows.

One of the two vascular features to be registered is called the *Template* and the other the *Input*. Thinning is performed for both the *Template* and the *Input* so that the resulting patterns consist of lines with one pixel width only. A sequential distance transformation (DT) is applied to create a distance map for the *Template* by propagation local distances [7]. The *Input* at different positions with respect to the corresponding transformations is superimposed on the *Template* distance map. A centreline mapping error (CME) to evaluate matching accuracy is defined as the average of feature points distance of the *Input* as follows:

$$CME = \frac{1}{N} \sum_{p(i,j) \in Input} DM_{Template}(p(i,j))^2 \tag{2}$$

$N$  is the total number of feature points in the *Input*,  $p(i,j)$  are the transformed positions of the original feature points in the *Input* and  $DM$  is the distance map created for the *Template* vascular features. It is obvious that a perfect match between the *Template* and *Input* images will result in a minimum value of CME.



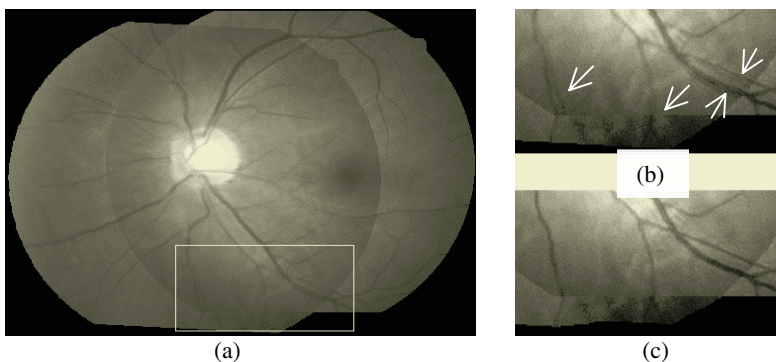
**Fig. 2.** (a) Vascular structure image at different levels from left to right: original resolution of 512x512 (level 0), intermediate resolution of 128x128 (level 2) and start resolution of 32x32 (level 4). (b) Distance image of (a) at the original level: the larger the distance the lighter the tone.

The search of minimum CME starts by using a number of combinations of initial model parameters. For each start point, the CME function are searched for neighboring positions in a sequential process by varying only one parameter at a time while keeping all the other parameters constant. If a smaller distance value is found, then the parameter value is updated and a new search of the possible neighbors with smaller distance continues. The algorithm stops after all its neighbors have been examined and there is no change in the distance measure. After all start points have been examined, transformations having local minima in CME larger than a prefixed threshold are selected as initial positions on the next level of finer resolution. The optimal position search of maximum similarity between tree structures is operated from coarse resolution towards fine resolution with less and less number of start points.

The final optimal match is determined by the transformation which has the smallest centreline mapping error at level 0 (the finest resolution). Once the relative parameters for the global transformation model have been computed, the registration between two retinal images is ready. The registration results could be examined by visually inspecting the constructed fusion maps of the original gray level images. Examples are illustrated in Figure 3(a).

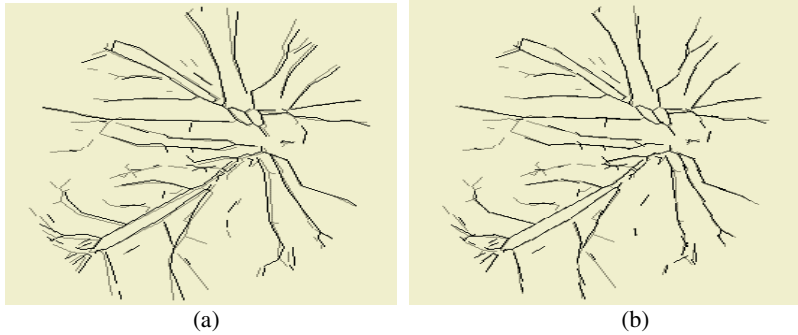
## 2.2 Local Elastic Matching

While the multi-resolution matching strategy is able to efficiently align retinal images globally, the alignment error caused by insufficient matching between some vessels remains unsatisfactory. The phenomenon of ‘ghost vessels’ that is more obvious around boundaries of overlapped region is caused by the assumption of the simple weak affine model used in globally matching two vascular features (see Figure 3(b)). In order to rectify the pitfall of the global transformation model, we propose a local elastic matching algorithm to further improve matching accuracy by eliminating the existence of ‘ghost vessels’ (Figure3(c)).



**Fig. 3.** A fusion map of two retinal fundus images formed by the computed rigid global transformation is shown in (a). The misaligned vessels (‘ghost vessel’) enclosed in the outlined frame in (a) have been clearly illustrated in (b) indicated by white arrows. (c) By applying the local elastic matching algorithm, near perfect alignment has been produced.

Let *Template* and *Input* be two binary vascular structures. Thinning is performed and single short lines and short branches are removed. Remaining lines and curves are approximated by fitting a set of straight lines. Each resulting straight line is then divided into smaller segments of approximately equal lengths referred as an 'element' which is represented by its slope and the position vector of its midpoint. Both vascular structures are, in turn, represented by a set of elements. Hence, the matching problem is equal to matching two sets of elements. Note that the number of elements in the two patterns need not be equal.



**Fig. 4.** (a) Overlapped image of *Input* pattern (black lines) and *Template* pattern (gray lines) before matching. (b) Overlapped image of patterns after matching.

The *Template* is elastically deformed in order to match the *Input* locally until the corresponding elements of both *Input* and *Template* meet, as illustrated in Figure 4. The objective is to achieve local alignment while to maintain the regional structure as much as possible. We elaborately create an energy function whose original format can be found in [8] to guide the deformation process.

$$\begin{aligned}
 E_1 = & -K_1^2 \sum_{i=1}^{N_I} \ln \sum_{j=1}^{N_T} \exp\left(-|\mathbf{T}_j - \mathbf{I}_i|^2 / 2K_1^2\right) f(\theta_{T_j, I_i}) \\
 & + \sum_{j=1}^{N_T} \sum_{k=1}^{N_T} w_{jk} (d_{T_j, T_k} - d_{T_j, T_k}^0)^2
 \end{aligned} \tag{3}$$

where  $N_T$  = number of *Template* elements,  $N_I$  = number of *Input* elements,  $\mathbf{T}_j$  = position vector of the midpoint of the  $j$ th *Template* element,  $\theta_{T_j}$  = direction of the  $j$ th *Template* element,  $\mathbf{I}_i$  = position vector of the midpoint of the  $i$ th *Input* element,  $\theta_{I_i}$  = direction of the  $i$ th *Input* element,  $\theta_{T_j, I_i}$  = angle between *Template* element  $T_j$  and *Input* element  $I_i$ , restricted within  $0-90^\circ$ ,  $f(\theta_{T_j, I_i}) = \max(\cos \theta_{T_j, I_i}, 0.1)$ ,  $d_{T_j, T_k}$  = current value of  $|\mathbf{T}_j - \mathbf{T}_k|$ ,  $d_{T_j, T_k}^0$  = initial value of  $|\mathbf{T}_j - \mathbf{T}_k|$ ,

$$w_{jk} = \frac{\exp\left(-|\mathbf{T}_j - \mathbf{T}_k|^2 / 2K_2^2\right)}{\sum_{n=1}^{N_T} \exp\left(-|\mathbf{T}_j - \mathbf{T}_n|^2 / 2K_2^2\right)}$$

$K_1$  and  $K_2$ : size parameters of the Gaussian windows which establish neighbourhoods of influence, and are decreased monotonically in successive iterations.

The first term of the energy function is a measure of the overall distance between elements of the two patterns. For each element  $\mathbf{I}_i$  of the *Input* pattern, the summation

$\ln \sum_{j=1}^{N_T} \exp(-|\mathbf{T}_j - \mathbf{I}_i|^2 / 2K_1^2) f(\theta_{T_j, I_i})$  is dominated by the contribution from the nearest

*Template* element  $\mathbf{T}_j$  with a similar slope. The value of the factor  $f(\theta_{T_j, I_i})$  is large for similar slopes and small for slopes perpendicular to each other. As the size  $K_1$  of the Gaussian window decreases monotonically in successive iterations, in order for the energy  $E_1$  to attain a minimum, each  $\mathbf{I}_i$  should have at least one  $\mathbf{T}_j$  attracted to it.

The second term is a weighted sum of all relative displacements between each *Template* element and its neighbors within the Gaussian weighted neighborhood of size parameter  $K_2$ . Minimization of this term minimizes the structural distortion of the *Template* pattern while each element is being moved. Each *Template* element normally does not move towards its nearest *Input* element but tends to follow the weighted mean movement of its neighbors in order to minimize the distortions within the neighborhood.  $K_2$  is initially large so that the large-scale distortions are kept small and the *Template* elements move collectively to align with the *Input* pattern in a coarse or global manner. As  $K_2$  is gradually decreased in successive iterations, finer and finer details of the two patterns are aligned.  $E_1$  is minimized by a gradient descent procedure. The movement  $\Delta \mathbf{T}_j$  applied to  $\mathbf{T}_j$  is equal to  $-\partial E_1 / \partial \mathbf{T}_j$  and is given by

$$\Delta \mathbf{T}_j = \alpha \sum_{i=1}^{N_I} u_{ij} (\mathbf{I}_i - \mathbf{T}_j) + 2\beta \sum_{m=1}^{N_T} (w_{mj} + w_{jm}) [(\mathbf{T}_m - \mathbf{T}_m^0) - (\mathbf{T}_j - \mathbf{T}_j^0)] \quad (4)$$

where

$$u_{ij} = \exp(-|\mathbf{I}_i - \mathbf{T}_j|^2 / 2K_1^2) f(\theta_{I_i, T_j}) / \sum_{n=1}^{N_T} \exp(-|\mathbf{I}_i - \mathbf{T}_n|^2 / 2K_1^2) f(\theta_{I_i, T_n})$$

$\mathbf{T}_j^0$  = initial value of  $\mathbf{T}_j$

Once we have identified correspondence of each feature points, to determine the displacement vectors of pixels in both the tree structure of the *Template* and the original retinal fundus image, the nearest  $N$  feature points of elements (middle point and two side-end points) of *Template* pattern are considered, where  $N$  is equal to 9 in our experiments. A normalized weighted sum of the displacement vectors of the feature points involved is used as the displacement vector of the pixel. Last, the registered positions of pixels of the *Template* retinal fundus image are ready to be computed in term of their global transformed positions and local displacement vectors. An example of the fused map is given in Figure 3(c).

### 3 Experiment Results

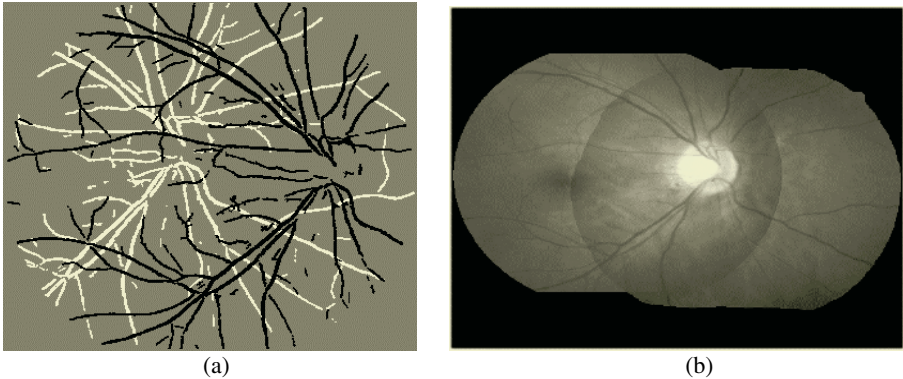
The image database that we use to evaluate the performance of the proposed registration algorithm consists of 115 gray level fundus images of both left and right eyes

from eleven patients. The image size is  $512 \times 512 \times 8$  bits. We randomly pair retinal fundus images captured at different times from the same eye of the same person resulting 268 pairs.

The depth for the fast multi-resolution matching method is set to 3, resulting in the size of the level 3 images being  $64 \times 64$  where vascular features can still be clearly recognized. We have 245 initial positions at the lowest resolution, namely:  $7 \times 7$  translation positions, separated by 5 pixels between -15 to 15 centred at the geometry centre of the  $64 \times 64$  frame, and 5 equidistant rotation angles. Given that the rotational movement of eyes is known to be small (less than 5 degrees), and taking into account the tilting of head during image capturing, the initial rotation positions are set at  $0$ ,  $\pm 3.5$  and  $\pm 7$  degree angles respectively.

The step-length for the translation parameters X or Y in vertical or horizontal coordinate directions is one pixel shift. The step-length for the rotation angle is related to the resolution of each computation level and should cause one pixel shift for at least one feature points (position change), normally the feature point with the largest distance from the original. Assuming the *Input* vascular structure spans from the one corner of the image frame over to another corner, the minimum rotating angle can be computed as  $\Delta\theta = 180 / (\sqrt{2} \times \pi \times width)$  degrees.

Taking into consideration the influence zone constraint, we tried two step-lengths:  $\Delta\theta$  and  $2 \times \Delta\theta$ . For example, in the experiments, at the starting level with the coarsest resolution of  $64 \times 64$ , step-lengths for rotational angle ( $\Delta\theta = 0.63$  degree of width=64) are 0.63 and 1.26 degrees respectively.



**Fig. 5.** (a) Overlapped vascular structures extracted from two gray level retinal fundus images, (b) Fusion map of the two retinal images formed by successful rigid global matching and local elastic alignment.

Initial parameter values used for calculation of movement vectors in the local elastic matching algorithm are carefully determined. Each line or curve is approximated by fitting a sequence of short straight lines ('elements') of about 20 pixels long in terms of the image size with which the local elastic matching begins ( $512 \times 512$ ).

Figure 5 shows an example of successful registration. The effectiveness and the efficiency of the algorithms are reflected in Table I. The average execution time taken for the entire matching algorithm is 207 seconds on a Pentium III Window XP com-

puter with 866MHz CPU and 384MB RAM. The average centreline mapping error is 1.85 pixels. It is clear that the employment of local elastic matching significantly improves the matching accuracy by reducing the average centreline mapping error achieved via global transformation down 0.94 pixels. The price is the extra 9 seconds of the average execution time.

**Table 1.** Computation complexity and Performance.

Algorithms	Average centreline mapping error (pixels)	Average time taken (s)
Global	2.79	198
Global+Local	1.85	207

## 4 Conclusion

In this paper, we have described how to apply a ‘global-to-local’ matching strategy to accurately align pairs of retinal fundus images with improved registration error. A multi-resolution global matching algorithm incorporated with a rigid model is employed to search for optimal parameters of translation and rotation. The computation complexity is low but local misalignments exist. In order to improve the registration accuracy, we adopt a local structure-deformed elastic matching algorithm to eliminate the existence of ‘ghost vessels’. Experiment results show the effectiveness and efficiency of the matching algorithms. Alignments are successfully achieved in 1.85 pixels of average centreline mapping errors and in 207 seconds of average execution time. The registration results have also been validated by visual inspection of the fusion maps.

## Acknowledgment

This research was supported by a grant FRG/04-05/II-15 from Hong Kong Baptist University, a grant (60403011) from National Natural Science Foundation of China, a grant (2003ABA012 ) from Hubei Provincial Science and Technology Department, and a grant (20045006071-17) from Wuhan government, China.

## References

1. D. E. Singer, D. M. Nathan, H. A. Fogel, A. P. Schachar, “Screening for diabetic retinopathy,” *Ann. Intern. Med.*, no. 116, pp. 660-671, 1992.
2. N. Ritter, R. Owens, J. Cooper, R. H. Eikelboom and P. P. V. Saarloos, “Registration of stereo and temporal images of the retina,” *IEEE Trans. Med. Imag.*, vol. 18, pp. 404-418, 1999.
3. A. Can, C.V. Stewart, B. Roysam and H.L. Tanenbaum, “A feature-based, robust, hierarchical algorithm for registering pairs of images of the curved human retina,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 24, no. 3, pp. 347-364, March 2002.

4. A. Pinz, S. Bernogger, P. Datlinger and A. Kruger, "Mapping the Human Retina," *IEEE Trans. Med. Imag.*, vol. 17, no. 4, pp. 606-619, Aug. 1998.
5. G. K. Matsopoulos, N. A. Mouravliansky, K. K. Delibasis, and K. S. Nikita, "Automatic retinal image registration scheme using global optimization techniques," *IEEE Trans. Info. Tech. Biomed.*, vol. 3, no. 1, pp. 47-60, Mar. 1999.
6. B. Fang, W. Hsu, M. L. Lee, "Reconstruction of vascular structures in retinal images," in *Proceedings. ICIIP'2003*, Barcelona, Spain, September 2003.
7. G. Borgefors, "Hierarchical Chamfer Matching: a parametric edge matching algorithm," *IEEE trans. Pattern Analysis Machine Intelligence*, vol. 10, no. 6, pp. 849-865, 1988.
8. C. H. Leung and C. Y. Suen, "Matching of Complex Patterns by Energy Minimization," *IEEE Transactions on Systems, Man and Cybernetics*, Part B, vol. 28, no. 5, pp. 712-720, 1998.

# A Model-Based Method for Face Shape Recovery

William A.P. Smith and Edwin R. Hancock

Department of Computer Science  
The University of York  
{wsmith, erh}@cs.york.ac.uk

**Abstract.** In this paper we describe a model-based method for recovering the 3D shape of faces using shape-from-shading. Using range-data, we learn a statistical model of the variation in surface normal direction for faces. This model uses the azimuthal equidistant projection to represent the distribution of surface normal directions. We fit the model to intensity data using constraints on the surface normal direction provided by Lambert's law. We illustrate the effectiveness of the method on real-world image data.

## 1 Introduction

Shape-from-shading provides an alluring yet somewhat elusive route to recovering 3D surface shape from single 2D intensity images. This has been partially motivated by psychological evidence of the role played by shape-from-shading in human face perception [1]. In addition, accurate recovery of facial shape would provide an illumination and viewpoint invariant description of facial appearance which may be used for recognition. Unfortunately, the method has proved ineffective in recovering realistic 3D face shape because of local convexity-concavity instability due to the bas-relief ambiguity. This is of course a well known effect which is responsible for a number of illusions, including Gregory's famous inverted mask [2]. The main problem is that the nose becomes imploded and the cheeks exaggerated. It is for this reason that methods such as photometric stereo [3] have proved to be more effective.

One way of overcoming this problem with single view shape-from-shading is to use domain specific constraints. Several authors [4–8] have shown that, at the expense of generality, the accuracy of recovered shape information can be greatly enhanced by restricting a shape-from-shading algorithm to a particular class of objects. For instance, both Prados and Faugeras [8] and Castelan and Hancock [7] use the location of singular points to enforce convexity on the recovered surface. Zhao and Chellappa [5], on the other hand, have introduced a geometric constraint which exploited the approximate bilateral symmetry of faces. This 'symmetric shape-from-shading' was used to correct for variation in illumination. They employed the technique for recognition by synthesis. However, the recovered surfaces were of insufficient quality to synthesise novel viewpoints. Moreover, the symmetry constraint is only applicable to frontal face images. Atick et al. [4] proposed a statistical shape-from-shading framework based on a low dimensional parameterisation of facial surfaces. Principal components analysis was used to derive a set of 'eigenheads' which compactly captures 3D facial shape.



Unfortunately, it is surface orientation and not depth which is conveyed by image intensity. Therefore, fitting the model to an image equates to a computationally expensive parameter search which attempts to minimise the error between the rendered surface and the observed intensity. This is similar to the approach adopted by Samaras and Metaxas [6] who incorporate reflectance constraints derived from shape-from-shading into a deformable face-model.

The aim in this paper is to construct a generic statistical model that can be used to capture the modes of variation in surface normal direction. We couple this model to the raw image brightness using the geometric shape-from-shading framework of Worthington and Hancock [9]. Unfortunately, the construction of such a model is not a straightforward task since the statistical representation of directional data has proved to be considerably more difficult than that for Cartesian data. Surface normals can be viewed as residing on a unit sphere and may be specified in terms of the elevation and azimuth angles. This representation makes the computation of distance difficult. For instance, if we consider a short walk across one of the poles of the unit sphere, then although the distance traversed is small, the change in azimuth angle is large. To overcome the problem, in this paper we draw on ideas from cartography. Our starting point is the *azimuthal equidistant* or Postel projection [10]. This projection has the important property that it preserves the distances between locations on the sphere. It is used in cartography for path planning tasks. Using the projection we transform the surface normals to points on a reference plane. We construct a statistical model of the surface normals using a standard point-distribution model on the tangent-plane.

We fit the model to 2D intensity images using ideas drawn from shape-from-shading. According to Worthington and Hancock [9], when the surface reflectance follows Lambert's law, then the surface normal is constrained to fall on a cone whose axis is in the light source direction and whose opening angle is the inverse cosine of the normalised image brightness. This method commences from an initial configuration in which the surface normals reside on the irradiance cone and point in the direction of the local image gradient. The statistical model is fitted to recover a revised estimate of the surface normal directions. The best-fit surface normals are projected onto the nearest location on the irradiance cones. This process is iterated to convergence, and the height map for the surface recovered by integrating the final field of surface normals.

## 2 Azimuthal Equidistant Projection

A needle map describes a surface  $z(x, y)$  as a set of local surface normals projected onto the view plane. Let  $\mathbf{n}_k(\mathbf{i}, \mathbf{j}) = (\mathbf{n}_k^x(\mathbf{i}, \mathbf{j}), \mathbf{n}_k^y(\mathbf{i}, \mathbf{j}), \mathbf{n}_k^z(\mathbf{i}, \mathbf{j}))^T$  be the unit surface normal at the pixel indexed  $(i, j)$  in the  $k^{th}$  training image. At the location  $(i, j)$ , the mean-surface normal direction is  $\hat{\mathbf{n}}(i, j) = \frac{\bar{\mathbf{n}}(i, j)}{\|\bar{\mathbf{n}}(i, j)\|}$  where  $\bar{\mathbf{n}}(i, j) = \frac{1}{K} \sum_{k=1}^K \mathbf{n}_k(i, j)$ . On the unit sphere, the surface normal  $\mathbf{n}_k(i, j)$  has elevation angle  $\theta_k(i, j) = \frac{\pi}{2} - \arcsin n_k^z(i, j)$  and azimuth angle  $\phi_k(i, j) = \arctan \frac{n_k^y(i, j)}{n_k^x(i, j)}$ , while the mean surface normal at the location  $(i, j)$  has elevation angles  $\hat{\theta}(i, j) = \arcsin \hat{n}^z(i, j)$  and azimuth angle  $\hat{\phi}(i, j) = \arctan \frac{\hat{n}^y(i, j)}{\hat{n}^x(i, j)}$ .

To construct the azimuthal equidistant projection we proceed as follows. We commence by constructing the tangent plane to the unit-sphere at the location correspond-

ing to the mean-surface normal. We establish a local co-ordinate system on this tangent plane. The origin is at the point of contact between the tangent plane and the unit sphere. The x-axis is aligned parallel to the local circle of latitude on the unit-sphere.

Under the equidistant azimuthal projection at the location  $(i, j)$ , the surface normal  $\mathbf{n}_k(i, j)$  maps to the point with co-ordinate vector  $\mathbf{v}_k(i, j) = (x_k(i, j), y_k(i, j))^T$ . The transformation equations between the unit-sphere and the tangent-plane co-ordinate systems are

$$x_k(i, j) = k' \cos \theta_k(i, j) \sin[\phi_k(i, j) - \hat{\phi}(i, j)]$$

$$y_k(i, j) = k' \left\{ \cos \hat{\theta}(i, j) \sin \phi_k(i, j) - \sin \hat{\theta}(i, j) \cos \theta_k(i, j) \cos[\phi_k(i, j) - \hat{\phi}(i, j)] \right\}$$

where  $\cos c = \sin \hat{\theta}(i, j) \sin \theta_k(i, j) + \cos \hat{\theta}(i, j) \cos \theta_k(i, j) \cos[\phi_k(i, j) - \hat{\phi}(i, j)]$  and  $k' = \frac{c}{\sin c}$ .

Thus, in Figure 1,  $CP'$  is made equal to the arc  $CP$  for all values of  $\theta$ . The projected position of  $P$ , namely  $P'$ , therefore lies at a distance  $\theta$  from the centre of projection and the direction of  $P'$  from the centre of the projection is true. The equations for the inverse transformation from the tangent plane to the unit-sphere are

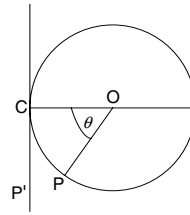


Fig. 1. The azimuthal equidistant projection

$$\theta_k(i, j) = \sin^{-1} \left\{ \cos c \sin \hat{\theta}(i, j) - \frac{1}{c} y_k(i, j) \sin c \cos \hat{\theta}(i, j) \right\}$$

$$\phi_k(i, j) = \hat{\phi}(i, j) + \tan^{-1} \psi(i, j)$$

where

$$\psi(i, j) = \begin{cases} \frac{x_k(i, j) \sin c}{c \cos \hat{\theta}(i, j) \cos c - y_k(i, j) \sin \hat{\theta}(i, j) \sin c} & \text{if } \hat{\theta}(i, j) \neq \pm \frac{\pi}{2} \\ -\frac{x_k(i, j)}{y_k(i, j)} & \text{if } \hat{\theta}(i, j) = \frac{\pi}{2} \\ \frac{x_k(i, j)}{y_k(i, j)} & \text{if } \hat{\theta}(i, j) = -\frac{\pi}{2} \end{cases}$$

and  $c = \sqrt{x^2 + y^2}$ .

### 3 Point Distribution Model

For each image location the transformed surface normals from the  $K$  different training images are concatenated and stacked to form two long-vectors of length  $K$ . For the pixel location indexed  $(i, j)$ , the first of these is the long vector with the transformed  $x$ -coordinates from the training images as components, i.e.  $\mathbf{V}_x(i, j) = (x_1(i, j), x_2(i, j), \dots, x_K(i, j))^T$  and the second long-vector has the  $y$  co-ordinate as its components, i.e.

$\mathbf{V}_y(i, j) = (y_1(i, j), y_2(i, j), \dots, y_K(i, j))^T$ . Since the azimuthal equidistant projection involves centering the local co-ordinate system (i.e. the mean direction is projected to the point  $(0, 0)$ ), the mean long-vectors over the training images are null. If the data is of dimensions  $M$  rows and  $N$  columns, then there are  $M \times N$  pairs of such long-vectors. The long vectors are ordered according to the raster scan (left-to-right and top-to-bottom) and are used as the columns of the  $K \times (2MN)$  data-matrix  $\mathbf{D} = (\mathbf{V}_x(1, 1)|\mathbf{V}_y(1, 1)|\mathbf{V}_x(1, 2)|\mathbf{V}_y(1, 2)| \dots |\mathbf{V}_x(M, N)|\mathbf{V}_y(M, N))$ . The covariance matrix for the long-vectors is the  $(2MN) \times (2MN)$  matrix  $\mathbf{L} = \frac{1}{K}\mathbf{D}^T\mathbf{D}$ . We follow Atick et al. [4] and use the numerically efficient method of Sirovich [11] to compute the eigenvectors of  $\mathbf{L}$ . Accordingly, we construct the matrix  $\hat{\mathbf{L}} = \frac{1}{K}\mathbf{D}\mathbf{D}^T$ . The eigenvectors  $\hat{\mathbf{e}}_i$  of  $\hat{\mathbf{L}}$  can be used to find the eigenvectors  $\mathbf{e}_i$  of  $\mathbf{L}$  using  $\mathbf{e}_i = \mathbf{D}^T\hat{\mathbf{e}}_i$ . We deform the azimuthal equidistant point projections in the directions defined by the matrix  $\mathbf{P} = (\mathbf{e}_1|\mathbf{e}_2| \dots |\mathbf{e}_K)$  formed from the leading  $K$  principal eigenvectors.

A vector of parameters  $\mathbf{b}$  describing a field of transformed surface normals on the local tangent plane  $\mathbf{v}_k$  is given by:  $\mathbf{b} = \mathbf{P}^T\mathbf{v}_k$ . To deform the field of surface normals, we can displace the transformed surface normals on the local tangent planes in the directions defined by the eigenvectors  $\mathbf{P}$ . The deformed field of surface normals can be transformed back onto the unit sphere using the inverse azimuthal equidistant projection equations given above.

## 4 Geometric Shape-from-Shading

If  $I$  is the measured image brightness, then according to Lambert's law  $I = \mathbf{n} \cdot \mathbf{s}$ , where  $\mathbf{s}$  is the light source direction. In general, the surface normal  $\mathbf{n}$  can not be recovered from a single brightness measurement since it has two degrees of freedom corresponding to the elevation and azimuth angles on the unit sphere. In the Worthington and Hancock [9] iterative shape-from-shading framework, data-closeness is ensured by constraining the recovered surface normal to lie on the reflectance cone whose axis is aligned with the light-source vector  $\mathbf{s}$  and whose opening angle is  $\arccos I$ . At each iteration the surface normal is free to move to an off-cone position subject to some smoothness or curvature consistency constraint. However, the hard constraint is re-imposed by rotating each surface normal back to its closest on-cone position. This process ensures that the recovered field of surface normals satisfies the image irradiance equation after every iteration.

Suppose that  $(\mathbf{n}')^l(i, j)$  is an off-cone surface normal at iteration  $k$  of the algorithm, then the update equation is  $\mathbf{n}^{l+1}(i, j) = \Theta(\mathbf{n}')^l(i, j)$  where  $\Theta$  is a rotation matrix computed from the apex angle  $\alpha$  and the angle between  $(\mathbf{n}')^l(i, j)$  and the light source direction  $\mathbf{s}$ . To restore the surface normal to the closest on-cone position it must be rotated by an angle  $\theta = \alpha - \arccos [(\mathbf{n}')^l(i, j) \cdot \mathbf{s}]$  about the axis  $(u, v, w)^T = (\mathbf{n}')^l(i, j) \times \mathbf{s}$ . Hence, the rotation matrix is

$$\Theta = \begin{pmatrix} c + u^2c' & -ws + wvc' & vs + wvc' \\ ws + wvc' & c + v^2c' & -us + vvc' \\ -vs + wvc' & us + vvc' & c + w^2c' \end{pmatrix}$$

where  $c = \cos(\theta)$ ,  $c' = 1 - c$  and  $s = \sin(\theta)$ .

The framework is initialised by placing the surface normals on their reflectance cones such that they are aligned in the direction opposite to that of the local image gradient. We use the irradiance cone constraint to fit our statistical model of surface normal variation to image brightness data.

## 5 Combining the Statistical Model and Geometric SFS

We train the statistical model using surface normals derived from range images of faces. The method could be trained on surface normal data delivered by shape-from-shading, but this is generally less reliable. To do this we used 200 range images of male and female subjects in frontal poses and neutral expressions [12]. Once trained, the statistical model represents the space of valid face shapes. By fitting the model to the data, we can extract the needle map within this shape space that is closest to a given field of surface normals. This “best fit” needle map is statistically constrained to represent a valid facial surface. The idea underpinning this paper is to fit the model to brightness images using the fields of surface normals estimated using the Worthington and Hancock shape-from-shading method. This is an iterative process in which we interleave the process of fitting the statistical model to the current field of estimated surface normals, and then re-enforcing the data-closeness constraint provided by Lambert’s law by mapping the surface normals back onto their reflectance cones. The algorithm can be summarised as follows:

1. Calculate an initial estimate of the field of surface normals  $\mathbf{n}$  by placing each normal on its reflectance cone in the direction of the negative local intensity gradient.
2. Each normal in the estimated field  $\mathbf{n}$  undergoes an azimuthal equidistant projection to give a vector of transformed coordinates  $\mathbf{v}$ .
3. The vector of best fit model parameters is  $\mathbf{b} = \mathbf{P}^T \mathbf{v}$ .
4. The vector of transformed coordinates corresponding to the best-fit parameters is  $\mathbf{v}' = (\mathbf{P}\mathbf{P}^T)\mathbf{v}$ .
5. Using the inverse azimuthal equidistant projection find  $\mathbf{n}'$  from  $\mathbf{v}'$ .
6. Find  $\mathbf{n}''$  by rotating  $\mathbf{n}'$  using  $\mathbf{n}''(i, j) = \Theta \mathbf{n}'(i, j)$ .
7. Stop if the difference between  $\mathbf{n}$  and  $\mathbf{n}''$  indicates convergence.
8. Make  $\mathbf{n} = \mathbf{n}''$  and return to step 2.

Upon convergence we output  $\mathbf{n}''$ , which satisfies the data-closeness constraint. However, given the variation in albedo in real world facial images, this may not be desirable. In this case we may choose to output  $\mathbf{n}'$  and an estimate of the albedo map. In other words we relax the data-closeness constraint at the final iteration and use the differences between observed and reconstructed image brightness to account for albedo variations. If the final best-fit field of surface normals is reilluminated using a Lambertian reflectance model, then the predicted image brightness is given by  $I(i, j) = \alpha(i, j)[\mathbf{s} \cdot \mathbf{n}'(i, j)]$  where  $\alpha(i, j)$  is the albedo at position  $(i, j)$ . Since  $I$ ,  $\mathbf{s}$ , and  $\mathbf{n}'$  are all known we can estimate the albedo at each pixel using the formula  $\alpha(i, j) = \frac{I(i, j)}{\mathbf{s} \cdot \mathbf{n}'(i, j)}$ . The combination of the final best-fit needle map and estimated albedo map allows for near photo-realistic reilluminations under novel illumination and viewpoint. In the next

section we demonstrate how both  $\mathbf{n}'$  and  $\mathbf{n}''$  are improvements over the needle maps estimated using the original curvature consistency constraints proposed by Worthington and Hancock [9].

## 6 Experiments

In this section we apply the algorithm described above to a number of real world face images. These images are drawn from the Yale B database [3] and are disjoint from the data used to train the statistical model. In the images the faces are in a frontal pose and were illuminated by a point light source situated approximately at the viewpoint, i.e. in direction  $[0\ 0\ 1]^T$ . We begin by analysing the behaviour of the algorithm over a number of iterations. We then show how the recovered needle map can be used to synthesise images of the input faces under novel illumination and from novel viewpoints.

In Figure 2 (left) we show the angular change as data-closeness is restored to the best fit needle map at the final iteration. From the plot it is clear that the changes are almost solely due to the variation in albedo at the eyes, eye-brows and lips. Aside from these regions there is very little change in surface normal direction, indicating the needle map has converged to a solution which satisfies the data-closeness constraint except in regions of actual variation in albedo. Using the technique described above, Figure 2 (right) shows the estimated albedo map. The results appear intuitively convincing. For instance, how the albedo map identifies the eyes, eyebrows, facial hair and lips. Moreover, there are no residual shading effects in the albedo map, and the nose is given constant albedo.



**Fig. 2.** Angular difference between final  $\mathbf{n}'$  and  $\mathbf{n}''$  (left) and estimated albedo map (right)

The algorithm converges rapidly, usually within 10 to 20 iterations. In fact, there is a considerable improvement in the needle map after only one iteration. This is because the statistical model provides a very strict constraint. The top row of Figure 3 shows how a needle map develops over 25 iterations of the algorithm. Since the needle maps satisfy data-closeness at every iteration, they would all appear identical when rendered with a light source from the original direction ( $[0\ 0\ 1]^T$ ). For this reason we show the needle maps reilluminated with a light source moved along the x-axis to subtend an angle of  $45^\circ$  with the viewing direction. After one iteration there is a significant global improvement in the recovered needle map. Subsequent iterations make more subtle improvements, helping to resolve convex/concave errors and sharpening defining features. For comparison the second row shows the corresponding needle maps recovered using the original curvature consistency constraint of Worthington and Hancock [9] reilluminated in the same manner. Although there is a steady improvement in the quality of the recovered normals, there are gross global errors as well as feature implosions around features such as the nose.

In Figure 3 we also show the surfaces recovered from the current best fit needle maps (third row) and the needle maps which satisfy data-closeness (bottom row) as

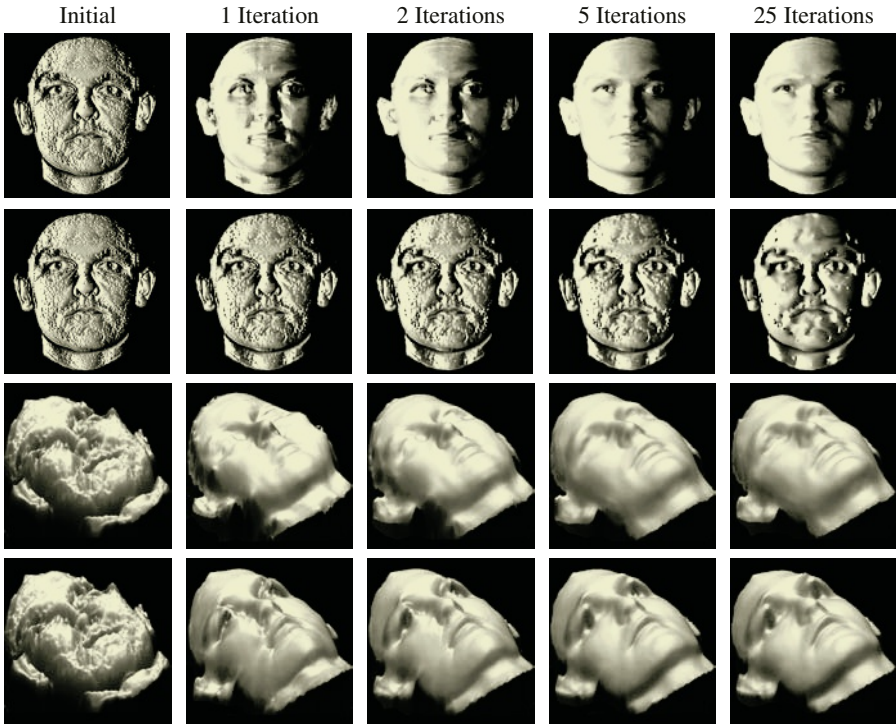


Fig. 3. Re-illuminated needle maps and recovered surfaces over 25 iterations of the algorithm

the algorithm iterates. Surface recovery is effected using the method of Frankot and Chellappa [13]. As one would expect, the imposition of data-closeness results in errors in the recovered surface where there is variation in albedo, most notably around the eyes and eye-brows. In both sets there is a clear improvement in the recovered surface as the algorithm iterates. The implosion of the nose is corrected, the surface becomes smoother and finer details become evident, for example around the lips.

Finally in Figure 4 we show how the estimated albedo maps and final best fit needle maps can be used to synthesise views of a face in novel pose and under novel illumination from a single image. In the first four columns the light source is moved to subtend an angle of  $45^\circ$  with the view direction along the positive and negative  $x$  and  $y$ -axes. In the fifth column the faces are shown rotated  $30^\circ$  about the vertical axis. The synthesised views are very convincing, even under large changes in lighting and viewpoint.

## 7 Conclusions

In this paper we have shown how to recover estimates of the 3D shape of faces from single frontal images. The method iterates between surface normal estimation using a geometrical shape-from-shading method and fitting a statistical model to the field of surface normals. This process can be posed as that of recovering the best-fit field



Fig. 4. Novel illumination and viewpoint

of surface normals from the statistical model, subject to constraints provided by the image irradiance equation. The method proves rapid to converge, and delivers realistic surfaces when the fields of surface normals are integrated. Our future plans revolve around placing the iterative process in a statistical setting using the EM algorithm and a von-Mises distribution to model the likelihood for the surface normal data. We also plan to develop ways of aligning the model with images which are not in a frontal pose.

## References

1. Johnston, A., Hill, H., Carman, N.: Recognising faces: effects of lighting direction, inversion, and brightness reversal. *Perception* **21** (1992) 365–375
2. Gregory, R.L.: Knowledge in perception and illusion. *Phil. Trans. R. Soc. Lond. B* **352** (1997) 1121–1128
3. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. PAMI* **23** (2001) 643–660
4. Atick, J.J., Griffin, P.A., Redlich, A.N.: Statistical approach to shape from shading: Reconstruction of 3D face surfaces from single 2D images. *Neural Comp.* **8** (1996) 1321–1340
5. Zhao, W.Y., Chellappa, R.: Illumination-insensitive face recognition using symmetric shape-from-shading. In: *Proc. CVPR.* (2000)
6. Samaras, D., Metaxas, D.: Illumination constraints in deformable models for shape and light direction estimation. *IEEE Trans. PAMI* **25** (2003) 247–264
7. Castelán, M., Hancock, E.R.: Acquiring height maps of faces from a single image. In: *Proc. 3DPVT.* (2004) 183–190

8. Prados, E., Faugeras, O.D.: Unifying approaches and removing unrealistic assumptions in shape from shading: Mathematics can help. In: Proc. ECCV. (2004) 141–154
9. Worthington, P.L., Hancock, E.R.: New constraints on data-closeness and needle map consistency for shape-from-shading. IEEE Trans. PAMI **21** (1999) 1250–1267
10. Snyder, J.P.: Map Projections—A Working Manual, U.S.G.S. Professional Paper 1395. United States Government Printing Office, Washington D.C. (1987)
11. Sirovich, L.: Turbulence and the dynamics of coherent structures. Quart. Applied Mathematics **XLV** (1987) 561–590
12. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Computer Graphics Proc. SIGGRAPH. (1999) 187–194
13. Frankot, R.T., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. IEEE Trans. PAMI **10** (1988) 439–451



# Visual Detection of Hexagonal Headed Bolts Using Method of Frames and Matching Pursuit

Pier Luigi Mazzeo, Ettore Stella, Nicola Ancona, and Arcangelo Distante

Istituto di studi sui sistemi intelligenti per l'automazione - C.N.R.  
Via G. Amendola, 122/D-I 70126 Bari, Italy

**Abstract.** In this paper we focus on the problem of automatically detecting the absence of the fastening bolts that secure the rails to the sleepers. The proposed visual inspection system uses images acquired from a digital line scan camera installed under a train. The general performances of the system, in terms of speed and detection rate, are mainly influenced by the adopted features for representing images and by their number. In this paper we use overcomplete dictionaries of waveforms, called frames, which allow dense and sparse representations of images and analyze the performances of the system with respect to the sparsity of the representation. Sparse means a representation with only few non vanishing components. In particular we show that, in the case of Gabor dictionaries, dense representations provide the highest detection rate. Moreover, the number of non vanishing components of 1% of the total reduces of 10% the detection rate of the system, indicating that very sparse representations do not heavily influence the performances. We show the adopted techniques by using images acquired in real experimental conditions.

## 1 Introduction

In the last years a large number of methods have been proposed by the computer vision community for facing the problem of visual inspection [1, 2]. This problem can be regarded as a particular instance of the most general problem of detecting objects in images as faces [3], pedestrians [4], balls [5] just for citing a few examples. Recently, such methods have been successfully applied for railway inspection and monitoring [6, 7]. In this field, the growing of the high-speed traffic on the rail tracks demands the development of sophisticated real-time visual inspection systems which are able to automatically detect rail defects. Usually, the maintenance of the railway plane is done by trained personnel who periodically observes the images recorded by a TV camera installed on a diagnostic coach. Actually, this manual inspection is lengthy, laborious and potentially hazardous and the results are strictly dependent on the capability of the observer to catch possible anomalies and recognize critical situations. The railway companies over the world are interested in developing automatic inspection systems which could increase the defect detection ability and decrease the inspection time in order to guarantee more frequently the maintenance of the entire railway network. In this context the detection of sleepers' anomalies, as well as missing fastening

elements, is an important task that an efficient inspection tool should supply. As described in our previous work [6] the Wavelet Transform has been successfully applied in railway context for the recognition of fastening elements. In other works [8, 9] this fastening element are recognized by using Independent Component Analysis (ICA) and Support Vector Machine (SVM). This kind of detection problem can be regarded as the problem of detecting flat objects from 2-D intensity image. Usually, such problems have been approached by using algorithms of edge detection, border following, thinning, straight line extraction, active contour (snake) following [10]. However, these methods fail if the patterns are distorted by imaging process, view-point changes, lighting changes or large intra-class variation among the patterns. In order to overcome these problems, the mostly used approaches in object recognition are based on feature extraction by a pre-processing technique. In this paper we focus on hexagonal-headed bolt images representations involving linear transformations of the original data with the main difference that we require the system functions not to be a basis. Such systems are, in general, constituted by much more elements than the ones present in a basis, and for this reason they are called *overcomplete* or *redundant* systems of functions. As described in Mallat [11] overcomplete dictionaries permit of representing signals in many different ways and it is possible to envisage that among all the possible representations, there is one suitable for a particular application. Sparsity is just one criteria for selecting a representation for a given image, the one with a few number of coefficients different from zero, particularly useful in the context of compression [12, 13]. In [14, 15] Matching Pursuit algorithms are employed in the classification context in particular Road Sign recognition and Face Identification. In this work we analyze sparse vs. no sparse representations of hexagonal headed bolt images, obtained by two different representation techniques, namely matching pursuit introduced by S. Mallat [11] and method of frames proposed by I. Daubechies [16]. In the case of MP is introduced a measure of sparsity called *sparseness factor* that is the ratio between number of non-zero coefficients and total number of atoms. We investigate these two methods by using overcomplete dictionary of Gabor functions, with different numbers of centers frequencies and orientations. Using this kind of dictionary, Method of Frame (MOF) achieves higher detection rate than Matching Pursuit (MP) in our context. However in MP case a sparseness factor less than 1% reduces only of 10% the performances of the whole system. This result indicates that sparse representations do not affect strongly the detection rate. This paper is organized as follows: In section 2 an overview of the Finite dimensional frame theory is presented. In section 3 we briefly describe MOF and the economic representation. The MP method is introduced in section 4. In section 5 is described the Gabor Dictionary employed. Finally experimental results are given in section 6.

## 2 Finite Dimensional Frames

In this section we analyze frames. We refer the reader to [17] for a review of the frame theory in generic Hilbert spaces. At this aim, consider a family of

vectors  $(\varphi_j)_{j=1}^\ell$  of  $\mathbb{R}^n$ . In some contexts [18], the family is called *dictionary* and the elements of the family are called *atoms*. By definition, the family of vectors  $(\varphi_j)_{j=1}^\ell$  constitutes a frame if there exist two constants  $A > 0$  and  $B < \infty$  such that, for all  $u \in \mathbb{R}^n$  we have:

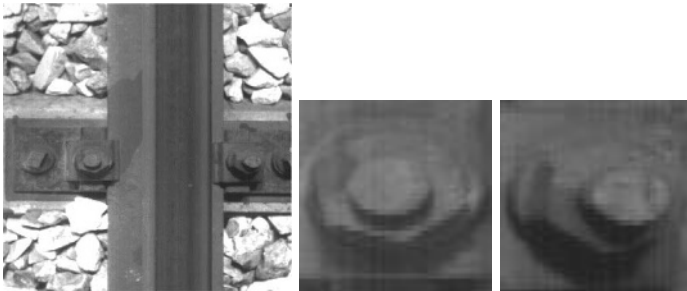
$$A\|u\|^2 \leq \sum_{j=1}^\ell (\langle u, \varphi_j \rangle)^2 \leq B\|u\|^2 \tag{1}$$

We call  $A$  and  $B$  *frame bounds*. Where we intend  $\langle a, b \rangle = a^\top b$ , if is not diversely specified. If  $A = B$  then we will say that  $(\varphi_j)_{j=1}^\ell$  is a *tight frame* and for all  $u \in \mathbb{R}^n$ :

$$u = \frac{1}{A} \sum_{j=1}^\ell \langle u, \varphi_j \rangle \varphi_j \tag{2}$$

Moreover, if  $A = 1$  then  $(\varphi_j)_{j=1}^\ell$  is an orthonormal basis.

We have a practical recipe for establishing if a system of vectors  $(\varphi_j)_{j=1}^\ell$  is a frame. In fact, build the matrix  $F$  having the vectors  $\varphi_j$  as rows. Compute the minimum and maximum eigenvalues of  $F^\top F$ . If  $\lambda_{min} > 0$  then the system  $(\varphi_j)_{j=1}^\ell$  is a frame, with frame bounds  $\lambda_{min}$  and  $\lambda_{max}$ .



**Fig. 1.** Images of rail fixed to the sleeper by hexagonal-headed bolts (left picture) and Sample image patterns of the hexagonal-headed bolts(right picture)

### 3 MOF and “Economic” Representations

Let  $F$  be a  $\ell \times n$  matrix having the frame vectors as its rows. The matrix  $F$  is the *frame operator*, where  $F : \mathbb{R}^n \rightarrow \mathbb{R}^\ell$ . Let  $c \in \mathbb{R}^\ell$  be the vector obtained when we apply the frame operator  $F$  to the vector  $u \in \mathbb{R}^n$ , that is:  $c = Fu$ . Then:

$$c_j = (Fu)_j = \langle u, \varphi_j \rangle \quad \text{for } j = 1, 2, \dots, \ell$$

Once we have computed the coefficients  $c$ , projecting the signal on the frame elements, there is a unique way to recover the signal  $u$ :

$$u = F^\dagger c \tag{3}$$

where  $F^\dagger = (F^\top F)^{-1}F^\top$  is the *pseudoinverse* of  $F$ .  $F$ , also called *analysis operator*, associates a vector of coefficients  $c$  (features) to a signal  $u$ , projecting the signal through the atoms of the dictionary. This operation involves a  $\ell \times n$  matrix.  $F^\dagger$ , also called *synthesis operator*, builds up a signal  $u$  as a superposition of the atoms of the dual dictionary weighted with coefficients  $c$ . Now the question is: among all possible representations of the signal  $u$  in terms of the frame elements, what properties have the coefficients  $c$ ? In [17] we show (see also [18]) that, among all representations of a signal in terms of the frame elements, MOF selects the one whose coefficients have minimum  $\ell^2$  norm.

## 4 Matching Pursuit

Matching pursuit [11] is a non linear algorithm that decomposes a signal into a linear expansion of waveforms that, in general, belong to a overcomplete dictionary of functions. It is an iterative procedure which, at each step, selects the atom of the dictionary which best reduces the residual between the current approximation of the signal and the signal itself. We analyze the method in the case of a signal  $u \in \mathbb{R}^n$ . Let  $(\varphi_j)_{j=1}^\ell$  be a frame of vectors of  $\mathbb{R}^n$ , with  $\|\varphi_j\| = 1$  for all  $j$ . At stage  $k = 0$ , let  $u^{(0)} = 0$  be the current approximation of the signal  $u$  with residual  $R^{(0)}$  given by  $R^{(0)} = u - u^{(0)}$ . The algorithm selects the dictionary atom that best correlates with the residual. In general, at stage  $k$  the algorithm builds the approximation  $u^{(k)}$  of  $u$  given by:

$$u^{(k)} = u^{(k-1)} + \left\langle R^{(k-1)}, \varphi_{j_k} \right\rangle \varphi_{j_k} \quad (4)$$

where:

$$j_k = \arg \max_{j \in J} \left| \left\langle R^{(k-1)}, \varphi_j \right\rangle \right| \quad (5)$$

with residual  $R^{(k)} = u - u^{(k)}$ . From the residual at the stage  $k$ -th we have:  $u = u^{(k)} + R^{(k)}$ . By using (4) recursively, at the stage  $k$  we have:

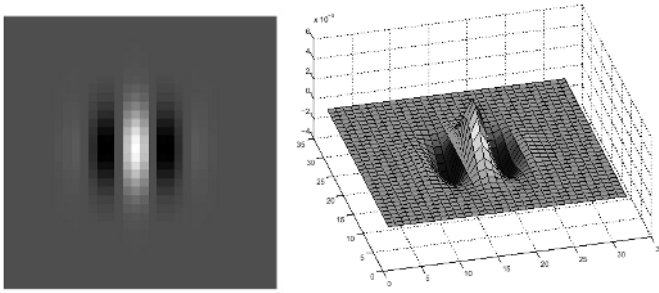
$$u = \sum_{i=1}^k \left\langle R^{(i-1)}, \varphi_{j_i} \right\rangle \varphi_{j_i} + R^{(k)} \quad (6)$$

From (6) follows that matching pursuit represents the signal  $u$  as linear combination of the dictionary atoms with coefficients computed minimizing at each step the residual. Some considerations are in order. In general the algorithm ends after a fixed number of iterations or when the residual is less than a given threshold. Matching pursuit is an example of a more general class of methods known as *greedy algorithms*: they do the best thing at every step. The algorithm provides a sparse representation of the signal.

## 5 Gabor Dictionary

The atoms used in this paper are Gabor atoms:

$$g(x, y) = \sqrt{2}K_s e^{-\pi(x^2+y^2)} e^{i\left(\frac{2\pi\omega_x}{N_x}x + \frac{2\pi\omega_y}{N_y}y\right)} \quad (7)$$



**Fig. 2.** Example of Gabor atom (picture on the left) and its 3-D representation (right picture)



**Fig. 3.** Experimental image acquisition setup

in their discrete version, so  $[x,y], [\omega_x, \omega_y] \in \mathbb{Z}^2$  and  $\vec{\omega} = [\omega_x, \omega_y]$  is fixed in this dictionary. The constant  $K_s$  normalizes the discrete norm of  $g$ . The choice of Gabor functions is due to the fact that they have optimal time-frequency resolution. The dictionary is defined by the set of Gabor atoms  $g_\gamma$ , with  $\gamma = (s, \mathbf{p}, \theta)$  and:

$s \in [0, \min(N_x, N_y)]$  scaling factor

$\mathbf{p} = [p_x, p_y]$  where  $p_x \in [0, N_x), p_y \in [0, N_y)$  translation

$\theta \in [0, \pi)$  rotation

with  $N_x \times N_y$  the size of the image,  $s$  the scaling factor that divides the spatial variables,  $\mathbf{p}$  the translation vector in  $x$  and  $y$  and  $\theta$  the angular resolution (rotation). As the application of these parameters is not commutative it is very important to fix the order a priori. The correct application order is:

1. Apply translation by  $[p_x, p_y] \in \mathbb{Z}^2$ .
2. Rotate by  $\theta$  the translated parameters.
3. Scale the translated and rotated variables by  $2^{\frac{s}{\text{NN}}}$ , where  $s$  is a discrete parameter  $s \in [0, \text{NN} \cdot \log_2(N)] \in \mathbb{Z}$  and  $\text{NN} \in [1, \log_2(N)] \in \mathbb{Z}$  ( $N$  is the size of the image).

We have chosen our parameters in such a way that the associated family is a frame of  $\mathbf{L}^2(\mathbb{R}^2)$  [20]. Intuitively,  $\Delta\theta$  and  $\Delta s$  must be small enough to allow some overlap between adjacent atoms in the Fourier domain. In [19] is demonstrated that this family of Gabor functions is complete.

## 6 Experimental Results

The images of the rail have been obtained by a line scan camera DALSA PI-RANHA 2 with 1024 pixels of resolution (maximum line rate of 67 kLine/s) with the transmission protocol Cameralink, installed under a diagnostic train during its maintenance route. Furthermore we have used the frame grabber PC-CAMLINK (Imaging Technology CORECO). In order to reduce the effects of variable natural lighting conditions, an appropriate illumination setup equipped with six OSRAM 41850 FL light sources has been installed too (see fig. 3). In this way the system should be robust against changes in the natural illumination.

Moreover, in order to synchronize data acquisition, a trigger is sent to the TV camera by the wheel encoder. The spatial resolution of the trigger is 3 mm. A pixel resolution of  $1 \times 1 \text{ mm}^2$  can be obtained choosing a TV camera with focal length of 12 mm. The integration time of the TV camera has been properly set in order to acquire images at maximum speed of 241 Km/h choosing the spatial resolution of 1 mm. A long video sequence of a rail network of about 5 Km has been acquired in order to experiment the proposed visual-based inspection system. Firstly, a number of sample images has been extracted from the sequence to create the training set for the neural classifiers. The remaining video sequence has been used to test the performance in term of detection rate e computational velocity of the developed inspection system. Positive and negative examples of the hexagonal-headed bolts have been manually extracted from the sequence training images (see fig. 1 right side). Each examples consist of a  $32 \times 32$  pixels subwindow where the width an height depend on the dimension of the hexagonal-headed bolts in the image (see fig. 1 left side). The training set is the same for all carried out experiments. This training set contains 301 positive examples and 301 negative examples of hexagonal headed bolts. The pre-processing strategies consist on Method of frames and Matching Pursuit with different residual error percentages. Both method use the Gabor Dictionary (see section 5). This atom dictionary is created with 4 central frequency 8 angular resolution and 1 octave. In this way our dictionary contains 32768 atoms. These different pre-processing techniques (described in detail in previous section) have been applied on the image examples. The neural classifiers Multi Layer Perceptron Network (MLPN) have been trained on this training set by back-propagation algorithm. In order to evaluate the generalization ability of the neural network classifier and the effects of the different pre-processing strategies on the images a test has been carried out on the validation set. This set contains 801 positive examples and 801 negative examples of hexagonal headed bolt. In table 1 the results of that test are shown. In the first column of Table 1 the pre-processing strategies are listed. The first row refers to MOF (Method of Frame), last three rows refer to MP (Matching Pursuit) method with crescent residual error percentages (up to 50%). In the second column of Table 1 the number of input coefficients for the classifiers are listed. The second column of the Table 1 contains the sparseness factor (the ratio between number of non-zero coefficients and total number of atom). Note that it is not present in the MOF row because this is a dense representation. In the last two columns of the same table the percentage of detection rate obtained from

**Table 1.** Detection rate for the hexagonal-headed bolt testing the system on images of the validation set

Pre-Processing	Number of input	Sparseness factor	Detection rate (%) of Back-propagation NN	
			TP	TN
MOF	32768	/	795/801 (99.2%)	787/801 (98.2%)
MP (10% Error)	32768	179/32768 (0.54%)	718/801 (89.6%)	709/801 (88.5%)
MP (30% Error)	32768	85/32768 (0.26%)	712/801 (88.8%)	725/801 (90.5%)
MP (50% Error)	32768	49/32768 (0.15%)	665/801 (83.0%)	725/801 (90.5%)

the test on the validation set is reported for MLPN classifier. Detection rates are given in terms of true positive (TP) and true negative (TN) rate.

## 7 Conclusion and Future Work

In this paper we have proposed a hexagonal bolt detection and recognition system in the railway maintenance context. MLPN classifiers was trained to recognize hexagonal-headed bolts. The images were pre-processed by using Method of Frames and Matching Pursuit techniques based on a Gabor Dictionary. The obtained trained networks were tested on a validation set to establish which pre-processing techniques perform better in terms of detection rate percentages. The results showed that MP technique with a decreasing residual error reach a good compromise between the sparseness factor and the detection percentages. The future work will be addressed to use the same techniques with different atoms dictionary.

## References

1. O. Silven, M. Niskanen, H. Kauppinen. Wood inspection with non-supervised clustering. *Machine Vision and Applications* (2003) Vol. 13 No.5-6: 275-285.
2. Y. Zhang, Z. Zhang, J. Zhang. Deformation visual of industrial parts with image sequence. *Machine Vision and Applications* (2004) Vol. 15 No.3 115-120.
3. K. Sung and T. Poggio. Example-based Learning for View-based Human Face Detection. Artificial Intelligence Laboratory. Massachusetts Institute of Technology, Cambridge, MA. 1994. A.I. Memo No. 1521.
4. C. Papageorgiou and T. Evgeniou and T. Poggio. A trainable pedestrian detection system. *Proceedings of Intelligent Vehicles*. Stuttgart, Germany, October, 1998.
5. N. Ancona and G. Cicirelli and E. Stella and A. Distanto. Ball detection in static images with Support Vector Machines for classification. *Image and Vision Computing*. Elsevier 2003. 21(8). 675-692.
6. Mazzeo P.L., Nitti M., Stella E. and Distanto A.: Visual recognition of fastening bolts for railroad maintenance. *Pattern Recognition Letters* Vol. **25** No. **6** (2004) 669-677.
7. C. Mandriota, M. Nitti, N. Ancona, E. Stella, A. Distanto, Filter-based feature selection for rail defect detection. *Machine Vision and Applications* (2004) 15: 179-185.

8. Mazzeo P.L., Ancona N., Stella E. and Distante A.: Visual Recognition of hexagonal headed bolts by comparing ICA to Wavelets. Proceedings of the IEEE Inter. Symposium on Intelligent Control-Houston(2003) 636–641
9. Mazzeo P.L., Ancona N., Stella E. and Distante A.: Fastening Bolts recognition in Railway images by Independent Component Analysis. Proc. 3rd Visualization, Imaging and Image Processing-Benalmadena(2003) 668–673
10. Andrade-Cetto J. and Kak Avinash C.: Object Recognition. Wiley Encyclopedia of Electrical Engineering. vol. Sup. 1 (2000) 449–470
11. Mallat S. and Zhang Z.: Matching Pursuit in a time-frequency. IEEE Transaction on signal processing **41**(1993) 3397–3415
12. Al-Shaykh O., Miloslavsky E., Nomura T., Neff R. and Zakhor. Video compression using matching pursuits. IEEE Transaction on Circuits and Systems for Video Technology. vol. **9** Iss. **1** (1999) 123–143
13. Bergeaud F. and Mallat S. Matching Pursuit of images. In Proceedings of the International Conference on Image Processing. (1995) 53–56
14. H.Hsu S. and L.Huang C. Road sign detection and recognition using matching pursuit method. Image and Vision Computing vol. **19** (2001) 119–129
15. Jonathon Phillips P. Matching Pursuit Filters applied to face Identification. IEEE Transactions on Image Processing vol. **7** No. **8** (1998) 1150–1164
16. Daubechies I. Ten lectures on wavelets. CBMS-NSF Regional Conferences Series in Applied Mathematics. SIAM, Philadelphia, PA (1992)
17. Ancona N. and Stella E. Image representations with overcomplete dictionaries for object detection. Technical Report R.I.- ISSIA/CNR-Nr. 02/2003. Istituto di studi sui sistemi intelligenti per l'Automazione - Consiglio Nazionale delle Ricerche, Bari, Italy, (2003)
18. Chen S., Donoho D., and Saunders M. Atomic Decomposition by basis pursuit. Technical Report 479, Dept of Statistics, Standford University (1995)
19. Tai Sing Lee. Image Representation Using 2D Gabor Wavelets. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **18** No. **10** (1996) 959–971
20. Mallat Stephane. A Wavelet Tour of Signal Processing. Academic Press, USA (1998)



# A New Region-Based Active Contour for Object Extraction Using Level Set Method

Lishui Cheng, Jie Yang, and Xian Fan

Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University  
200030, Shanghai, P.R. China  
{lishuicheng, jieyang, yivanne}@sjtu.edu.cn

**Abstract.** Object extraction or image segmentation is a basic problem in image analysis and computer vision. It has been dealt with in various forms. Variational method is an emerging framework to tackle such problems where the aim is to create an image partition that follows the data while at the same time preserving certain regularity. In this paper, we propose a new energy functional which is based on the region information of an image. The region-based force makes our variational flow robust to noise and provides a global segmentation criterion. Furthermore, our method is implemented using level set theory, which makes it easy to deal with topological changes. Finally, in order to simultaneously segment a number of different objects in an image, a hierarchical method is presented.

## 1 Introduction

Object extraction is a very popular low-level topic of research in image processing and computer vision with its applications to remote sensing, medical imaging and video tracking. Active contour model, since it was first proposed in [1], has been extensively studied in this area. The central idea behind active contour model is to evolve a curve or surface based on energy minimization method under the influence of image dependent forces, regularity constraints and certain user-specified constraints.

Originally, active contours are boundary-based methods, which usually need an edge detector to stop the evolving curve on the boundaries of the desired objects. Snakes [1], balloons [2] and recently the geodesic active contour [3] are all driven towards to the edges of an image. However, such methods only use the local information on the boundary of an object, which makes them sensitive to noise.

Recently, there has been a great interest in region-based active contours. In [4], Chan and Vese proposed a region-based active contour which was derived from Mumford-Shah functional [5]. The main idea behind Chan-Vese model is to use piecewise constant functions which are represented by the intensity means of different regions to approximate the original image with some regularity terms. In [6], Tsai et al. also proposed a curve evolution method based on the Mumford-Shah functional.

A third form of active contours is to unify boundary and region-based segmentation approaches. In [7], Zhu and Yuille proposed a statistical variational approach which combined the geometrical features of a snake/balloon model and the statistical techniques of region growing. N. Paragios also proposed a geodesic active region

model which was implemented in level set method [8]. We also note that recently X. Xie and M. Mirmehdi have proposed a region-aided geometric snake [9].

In this paper, we propose a new energy functional which is totally based on region features of an image. Compared with traditional boundary based active contours, our method is more robust to noise. Furthermore, we implement our method in the level set framework, thus it can deal with topological changes automatically and be extended to higher dimensions easily. Finally, unlike other active contours in [3] and [9], our contour is more insensitive to the initialization due to the region force. It can be placed near or far away from the boundaries of object.

Our work can be seen as an extension of Chan-Vese model. However, unlike their Mumford-Shah functional-based method, our contour is based on a new proposed functional and the corresponding evolution equation is more numerically stable. What's more, in order to segment multiple objects, we introduce a hierarchical method which was also used in Tsai et al. [6].

The remainder of this paper is organized as follows. Section 2 introduces the Chan-Vese model. Section 3 describes our variational active contour in detail. Section 4 presents the experimental results. Conclusions and future work are given in Section 5.

## 2 Related Work

In this section, we summarize the active contour model for segmenting bimodal images developed by Chan and Vese [4].

In the variational framework, an image  $I_0$  is usually considered a real-valued bounded function defined on  $\overline{\Omega}$ , where  $\Omega$  is a bounded and open subset of  $R^2$  (in two dimension case) with  $\partial\Omega$  as its boundary.

According to level set theory originally proposed by Osher and Sethian in [10], a geometric active contour can be represented by the zero level set of a real-valued function  $\phi: \Omega \subset R^2 \rightarrow R$  which evolves in an image  $I_0$  according to a variational flow in order to segment the object from the image background. Since it was proposed, level set theory has made great success in image processing community [11]. Some of the biggest advantages of level set method are as follows. Firstly, unlike traditional parametric active contour models [1], level set based active contours are parametric-independent and hence can deal with topological changes naturally. Secondly, level set contours can be extended to three and higher dimensions, which is needed in many image processing applications. Thirdly, level set method usually has mature numerical implementation [11].

The active contour model developed in [4] is based on Mumford-Shah functional and level set theory. The main idea is to minimize the following "fitting" energy functional with a length regularization term:

$$\begin{aligned}
 F(\phi, c_1, c_2) = & \mu \int_{\Omega} \delta(\phi) |\nabla \phi| dx dy + \lambda_1 \int_{\Omega} |I_0 - c_1|^2 H(\phi) dx dy \\
 & + \lambda_2 \int_{\Omega} |I_0 - c_2|^2 (1 - H(\phi)) dx dy
 \end{aligned} \tag{1}$$

where  $\mu, \lambda_1, \lambda_2$  are the scaling parameters,  $c_1$  and  $c_2$  are the mean intensities inside and outside the active contour  $\Gamma$  (see Fig. 1) defined as follows:

$$c_1 = \frac{\int_{\Omega} I_0(x, y)H(\phi(x, y))dxdy}{\int_{\Omega} H(\phi(x, y))dxdy} \tag{2}$$

$$c_2 = \frac{\int_{\Omega} I_0(x, y)(1 - H(\phi(x, y)))dxdy}{\int_{\Omega} (1 - H(\phi(x, y)))dxdy} \tag{3}$$

In the numerical implementation,  $H(\phi)$  is the regularized Heaviside function defined as:

$$H_{\epsilon}(z) = \frac{1}{2} \left( 1 + \frac{2}{\pi} \arctan\left(\frac{z}{\epsilon}\right) \right) \tag{4}$$

Similarly,  $\delta(\phi)$  is the regularized version of Delta function defined as:

$$\delta_{\epsilon}(z) = \frac{1}{\pi} \bullet \frac{\epsilon}{\epsilon^2 + z^2} \tag{5}$$

Note that when  $\epsilon \rightarrow 0$ , both regularized versions converge to standard Heaviside function and Delta function.

Using the gradient decent method, Chan and Vese derived their evolution equation as follows:

$$\phi_t = \delta_{\epsilon}(\phi) \left[ \mu \nabla \bullet \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (I_0 - c_1)^2 + \lambda_2 (I_0 - c_2)^2 \right]$$

(6)

### 3 Proposed Method

In this section, we will describe the proposed variational flow.

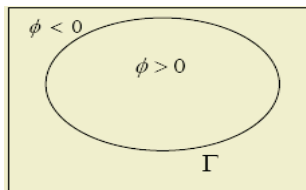


Fig. 1. The domain  $\Omega$  divided into two regions by the curve  $\Gamma$  where  $\phi = 0$

From Chan-Vese model, we notice that a bimodal image can be approximated by the following equation which we call piecewise constant image function:

$$I = c_1 \times H(\phi) + c_2 \times (1 - H(\phi)) \tag{7}$$

Based on this observation, we propose a new energy functional whose main idea is to minimize the difference between the piecewise constant image function and the original image while keeping some regularity. The proposed functional is as follows:

$$F(\phi, c_1, c_2) = \int_{\Omega} \frac{1}{2} (I - I_0)^2 dx dy + \mu \int_{\Omega} |\nabla \phi|^2 dx dy \tag{8}$$

where the first term is a fidelity term, and the second term is a Tikhonov regularization term.  $\phi$ ,  $c_1$  and  $c_2$  are defined as in Chan-Vese model,  $\mu$  is a regularizing parameter. The purpose of using the regularization term is to keep the level curve smooth and regular. Moreover, since our regularization term is imposed on the whole image domain, the evolution of all the level sets in our method is meaningful. So we need few reinitialization of level set function, which can reduce the computational complexity.

Using the fundamental lemma of calculus of variations, minimizing  $F$  with respect to  $\phi$ , we get the corresponding Euler-Lagrange equation:

$$\frac{\partial F}{\partial \phi} = \delta(\phi)(I - I_0)(c_1 - c_2) - \mu \Delta \phi \tag{9}$$

Next using the steepest descent method, we get the evolution equation as follows:

$$\frac{\partial \phi}{\partial t} = \delta(\phi)(I - I_0)(c_2 - c_1) + \mu \Delta \phi \tag{10}$$

Compared to Chan-Vese evolution equation, the  $\delta(\phi)$  in the regularization term is replaced by “1” in our method. For the purpose of numerical calculation, the  $\delta$  function tends to create oscillations if  $\epsilon$  in the regularized version is too small, while if it is too large, the accuracy of numerical method can decrease. Therefore our variational flow has a better numerical stability. Furthermore, because we use a Tikhonov regularization term which is a very strict regularization term, our method is more robust to noise than Chan-Vese model in some cases.

*Hierarchical Method:* In order to extract more than one object from an image, we adopt a hierarchical method. For a certain image, we firstly apply the active contour model proposed in previous section. At the end of this step, we get a contour which gives the boundaries of the segmented object. If there is a need for further segmentation, we then select one of the regions generated from the previous stage and apply our algorithm again only to this region. Therefore we can always only segment interesting areas.

*Implementation Remarks:* In numerical implementation,  $c_1$  and  $c_2$  are computed by the standard Heaviside function.  $I$  is calculated by the regularized version of Heaviside function.  $\delta(\phi)$  is computed by the regularized version of Delta function.

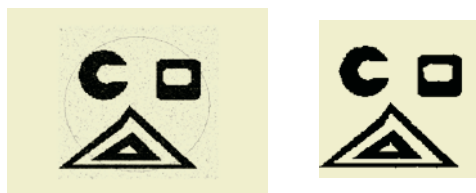
## 4 Experimental Results

In this section, we will give the experimental results obtained by using the new active contour model described in previous section. The segmented result is represented by a binary image. According to our algorithm, the image area where the level set function

$\phi > 0$  is the extracted object and  $\phi < 0$  is the background, so we directly use gray level “1” (or “0”) to indicate the object and “0”(or “1”) to indicate the corresponding background without using any post processing step.

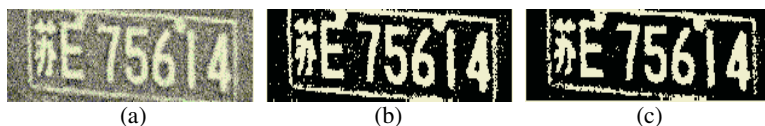
Our experiments were performed on a Celeron 633 MHz PC. In the numerical experiments, we generally choose the parameters as follows:  $\mu = 1$ ,  $\varepsilon = 1$ , the time step is 0.1, and the space step is 1.

In fig. 2, we apply our method on a synthetic image, with Gaussian noise, of which the mean is zero and the variance is 0.02. We note that this image has multifarious shapes and interior contours. The result indicates that our method has the capability to extract object from noisy image even with complex topology due to the region forces and level set technique used in our geometric flow. Furthermore, it can also extract interior boundaries of object.



**Fig. 2.** Result on a synthetic noise image

We show the segmentation results on a real car license plate image with noise in Fig. 3. We compare it with the results obtained by applying Chan-Vese model. The experiments demonstrate that in this case our method is more robust to noise than Chan-Vese model as discussed in section 3.



**Fig. 3.** Results on a car plate image with noise: (a) original image, (b) Chan-Vese model and (c) proposed method

We end our experiments by using a medical image in Fig. 4. The image is downloaded from the Brainweb [12], with T1-weighted, slice thickness of 1 mm, intensity inhomogeneity of 20%, and noise level of 7%. In this experiment the time step is 0.02. The MRI data is pre-processed to extract the region of interest by using the software MRICro [13]. In this case, there are four parts in an image: gray matter, white matter, cerebrospinal fluid and back ground. So we use our hierarchical method to extract the gray matter and the white matter separately. The experiments indicate that our variational flow gives good results under the noise level.

## 5 Conclusion and Future Work

In this paper, we propose a new variational active contour model to extract object from an image using level set method. Our variational flow is based on the region forces derived from the image and thus robust to noise. Compared to traditional boundary-based active contours, our contour is more insensitive to initialization. Moreover, our contour needs few reinitializations during evolution process. Finally, In order to extract multiple objects, we introduced a hierarchical method. The experiments demonstrate that our method is very effective.



**Fig. 4.** Results on a MRI brain image: (a) is the original image, (b) is the extracted white matter, and (c) is the extracted gray matter

Since our method is implemented in level set framework, it is easily extended to three dimensions. This is very useful especially in medical applications. Our method can also be used with other boundary-based active contour models, for example, geodesic active contour [3]. These topics will be the subject of future work.

## References

1. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1988
2. L. Cohen, "On active models and balloons," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 53, pp. 211-218, 1991
3. V. Caselles, R. Kimmel, and G. Sapiro, "On geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61-79, 1997
4. T. Chan, and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266-277, 2001
5. D. Mumford, and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communication on Pure and Applied Mathematics*, vol. 42, pp. 577-684, 1989
6. A. Tsai, A. Yezzi, A. Willsky, "Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1169-1186, 2001
7. S. Zhu, and A. Yuille, "Region competition: unifying snakes, region growing and bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884-900, 1996
8. N. Paragios, and R. Deriche, "Geodesic active regions: a new paradigm to deal with frame partition problems in computer vision," *Journal of Visual Communication and Image Representation*, vol. 13, pp. 249-268, 2002

9. X. Xie, and M. Mirmehdi, "RAGS: region-aided geometric snake," IEEE Transactions on Image Processing, vol. 13, no. 5, pp. 640-652, 2004
10. S. Osher, and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," Journal of Computational Physics, vol. 79, pp. 12-49, 1988
11. Y.-H. Tsai, and S. Osher, "Level set methods in image science", Image Processing, 2003. Proceedings.2003 International Conference on Volume: 2, 14-17 Sept. 2003 Pages:II - 631-4 vol.3
12. C.A. Cocosco, V. Kollokian, R. K.-S. Kwan, A. C. Evans, "Brain Web: Online Interface to a 3D MRI Simulated Brain Database," NeuroImage, vol. 5, no. 4, part 2/4, S425, 1997 -- Proceedings of 3-rd International Conference on Functional Mapping of the Human Brain, Copenhagen, May 1997, <http://www.bic.mni.mcgill.ca/brainweb/>.
13. <http://www.psychology.nottingham.ac.uk/staff/cr1/mricro.html>

# Improving ASM Search Using Mixture Models for Grey-Level Profiles

Yanong Zhu<sup>1</sup>, Mark Fisher<sup>1</sup>, and Reyer Zwiggelaar<sup>2</sup>

<sup>1</sup> School of Computing Sciences, University of East Anglia  
Norwich, NR4 7TJ, UK

{yz, mhf}@cmp.uea.ac.uk

<sup>2</sup> Department of Computer Science, University of Wales  
Aberystwyth, Ceredigion, SY23 3DB, UK  
rrz@aber.ac.uk

**Abstract.** The use of Active Shape Models (ASM) has been shown to be an efficient approach to image interpretation and pattern recognition. In ASM, grey-level profiles at landmarks are modelled as a Gaussian distribution. Mahalanobis distance from a sample profile to the model mean is used to locate the best position of a given landmark during ASM search. We present an improved ASM methodology, in which the profiles are modelled as a mixture of Gaussians, and the probability that a sample is from the distribution is calculated using the probability density function (pdf) of the mixture model. Both improved and original ASM methods were tested on synthetic and real data. The performance comparison demonstrates that the improved ASM method is more generic and robust than the original approach.

## 1 Introduction

Active Shape Modelling (ASM) [3] has been applied in many image analysis applications, such as medical image analysis, facial recognition and video object tracking, mainly due to its capability to deal with the variation of both shape and the signal intensity of the target object [1]. Image segmentation using the conventional ASM method can be divided into two stages. The first is the modelling (or training) stage, in which a parameterised statistical shape model is built from labelled training images. Grey-level profiles normal to the object boundary at each landmark are modelled as a single Gaussian distribution. At the second stage, a shape instance is deformed in accordance with the model to search for a boundary which optimally segments the object. This is an iterative optimisation process consists of two major steps: (a) search for better positions for each individual landmark using grey level statistical models, and (b) fit the shape model to new landmarks by updating the shape model parameters. Step (a) ensures that the boundary is placed at a location where the image structure around the boundary or within the object is most similar to that is modelled from the training data, and step (b) ensures that the segmentation can only produce plausible shapes. Both steps are crucial to the final search results.

A collection of practical improvements are found in the literature. Most of the improvements were aimed at shape variation modelling and generation of shape instances.



Cootes et al. [2] used the Gaussian mixtures to model the distribution of the landmarks and hence the shape variation. Rogers and Graham [5] improved ASM by using M-estimator and random sampling approaches to robust parameter estimation instead of Gaussian distribution based estimation. Twining et al. [6] described the use of Kernel Principal Component Analysis (KPCA) to model the variability in a class of shapes. On the other hand, van Ginneken et al. [7] used optimal image features instead of grey-level profiles for ASM search, and applied a  $k$ -Nearest Neighbour ( $k$ NN) classifier to find the displacement of landmarks. These improvements have achieved credible performance and largely increased the efficiency and robustness of ASM methods.

We have applied ASM to segment objects of interest from image data sets. As shown by some of the experiments, when the grey-level variation around the object border is too complex to be modelled as a single Gaussian distribution, the use of Mahalanobis distance to measure the distance from a sample to the mean of the distribution becomes inaccurate and consequently causes invalid search results. In this case, a more accurate representation of the image structure variation is expected to improve the performance of ASM boundary search.

In this paper, we concentrate on the modelling aspects of grey-level profiles, as well as locating of landmark positions using the profile models. Instead of a single Gaussian distribution and Mahalanobis distance, we use a Gaussian mixture to model the profiles, and the probability that a sample profile comes from the distribution is measured by the total probability of sub-distributions.

## 2 Methods

The key issue in our improved ASM method is that the grey-level profiles are no longer treated as a single Gaussian distribution, but as a mixture of Gaussians. The grey-level profiles, rather than their derivatives, are used to model the intensity variation so that original image information can be preserved. We assume that the sum of a certain number of Gaussian distributions can represent the distribution of these profiles. The Expectation Maximization (EM) algorithm [4] is applied to obtain the optimal Gaussian mixture. The probability that a sample profile is from the population is calculated by the combination of the probabilities that it belongs to each of the mixture components.

### 2.1 Profile Modelling Using Finite Gaussian Mixture Models

Finite mixture modelling is a powerful tool for density estimation and can be regarded as a flexible way to represent a probability density function (pdf). At the ASM modelling stage, intensity profiles are extracted from all training images at every landmark. For each landmark these profiles are treated as a set of samples in  $\mathbb{R}^d$ , denoted as  $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$ , where  $N$  is the number of images in the training set (i.e. the number of profiles for each landmark). A mixture of  $M$  simple distributions (e.g. Gaussians) can be used to represent the underlying distribution of such a set of profiles. The pdf of a sample profile  $x_i$  can be written as

$$p(x_i|\Theta) = \sum_{j=1}^M \alpha_j p_j(x_i|\mu_j, \sigma_j) \quad (1)$$

where  $\alpha_j$  is the mixing proportion of each component with  $\sum_{j=1}^M \alpha_j = 1$ ,  $0 \leq \alpha_j \leq 1$ , and  $p_j$  is the component density function parameterized by  $(\mu_j, \sigma_j)$  (respectively the mean and standard deviation for the Gaussian distributions), and  $\Theta = (\alpha_1, \alpha_2, \dots, \alpha_M, \mu_1, \sigma_1, \mu_2, \sigma_2, \dots, \mu_M, \sigma_M)$  are the model parameters to be estimated.

To obtain the optimal Gaussian mixture model, we used the EM algorithm to estimate the number of model components and the parameters of each component. Some pre-conditions must be fixed before the EM algorithm is applied to parameter estimation: (a) the functional form of each component pdf, and (b) the number of components  $M$ . In our case, the choice of component pdf is Gaussian, so we only search over the component number for the overall maximum log-likelihood.

## 2.2 Improving ASM Search

The intensity profiles for landmark  $P_i$  are modelled as a Gaussian mixture distribution characterised by the number of components  $M_i$  and parameters  $\Theta_i$ , where  $i$  is the index of the landmark. During search, an initial shape is placed on the target image. The region around each landmark is checked to find a optimal position for this landmark. The optimal position,  $\hat{P}_i$ , is the location where the local profile has maximum probability as determined by the mixture distribution,

$$\hat{P}_i \leftarrow \operatorname{argmax}_P p(x_P | \Theta_i) = \operatorname{argmax}_P \sum_{j=1}^{M_i} \alpha_{ij} p_j(x_P | \mu_{ij}, \sigma_{ij}) \quad (2)$$

where  $x_P$  is the intensity profile at position  $P$ , and  $P$  is selected along the profile across the current landmark  $P_i$ .

## 3 Experiments and Results

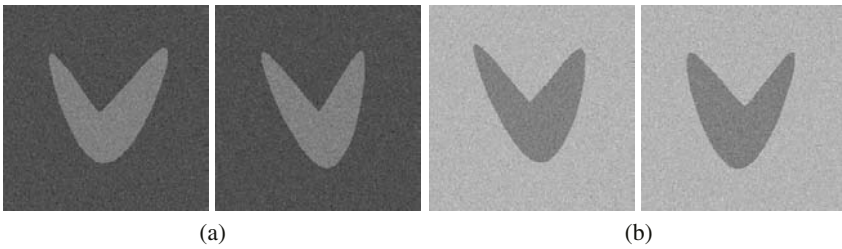
The improved ASM methodology was applied to two different sets of data, synthetic images and hand video images. The main aim of these experiments is to demonstrate the strength of the proposed method when applied to images with simple and complex intensity variations.

### 3.1 Synthetic Images

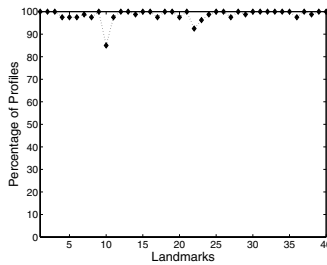
The use of synthetic images in the testing of a method enables us to compare the results of the method with *real* ground truth. In our experiments, the synthetic image data set,  $S$ , consists of two subsets  $S_A$  and  $S_B$ , each of which includes 40  $256 \times 256$  images. A ‘V’-shape target object with additive shape variation is placed in the center of the images. Four key points are used to generate a nonuniform rational B-spline curve that represents the ‘V’-shape boundary. To produce the shape variation, a displacement is added to each of these key points when the image is created. The displacement,  $(dx, dy)$ , is randomly selected from a Gaussian distribution, with the standard deviation of 6 pixels. Nine points are evenly chosen on each of the four segments on the curve. Hence, 40 landmarks are used to represent the target object. The grey level intensity of the target

is 128, while the background grey level for  $S_A$  and  $S_B$  are 80 and 180, respectively. Finally, we add Gaussian noise, with standard deviation of 16, to the images to simulate more realistic images. Set  $S$  contains all 80 images, and as such two different intensity distributions. Example images from each subset are given in Fig. 1.

We first investigate the number of Gaussian components used to model the intensity profiles at each landmark. Fig. 2 shows the percentage of profiles that are modelled by a mixture of two Gaussians, and for all but two landmarks, more than 95% of the profiles are modelled by these Gaussians. Since the two subsets of synthetic images are distinct, in most cases the intensity profiles are modelled as two major Gaussians plus a small third one. Such modelling provides a more accurate representation of the intensity variation than a single Gaussian model as used in the original ASM method.

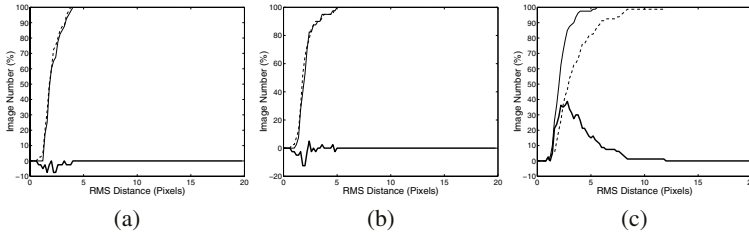


**Fig. 1.** Two example images from (a) set  $S_A$  and (b) set  $S_B$ .



**Fig. 2.** Percentage of profiles at each landmark that belong to the main two classes.

Secondly, leave-one-image-out experiments were performed on all the three sets respectively. Both original and improved ASM methods were applied to segment the 'V'-shape from the images. Root-Mean-Square Distance (RMSD) is commonly used to measure the similarity between two shapes. Therefore segmentation methods can be evaluated by comparing the RMSD from the segmentation results to the ground truth. Furthermore, the distribution of RMSD, i.e. the number of images against RMSD value, can provide a statistical comparison over a large data set. The RMSD distributions of segmentation results for synthetic images are shown in Fig. 3. These results indicate equivalent performance for the separate sets of images, but a significant improvement when the improved ASM method is applied to the complete data set.



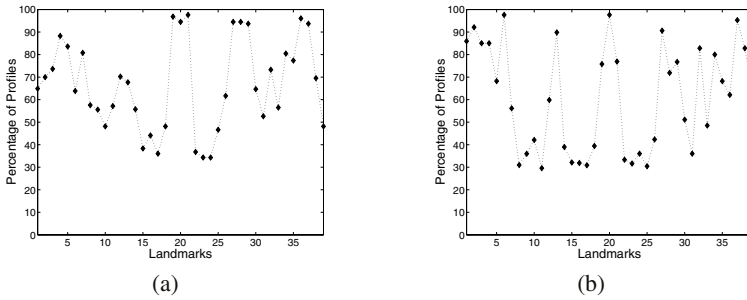
**Fig. 3.** Results on synthetic data. The distribution of RMSD, between the ground-truth and original ASM (dotted lines), the ground-truth and improved ASM (thin solid lines), the difference in performance between the improved and original ASM (thick solid lines). Here (a) set  $S_A$ , (b) set  $S_B$  and (c) set  $S$ .

### 3.2 Hand Video Images

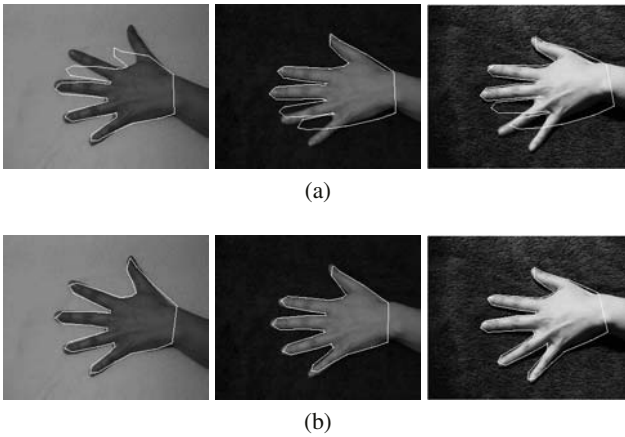
Subsequently, the improved ASM method was applied to a set of hand images. Three video clips of a volunteer's left hand were taken under different background and lighting conditions. The first video was taken on a white background without additional lighting. A dark blue background was used for the other two while an additional spot light was used for the third video clip. The videos were captured using a Fuji S304 digital camera and have a significant noise level. The finger movements composed the shape variation in these clips. Respectively from these three clips, 40, 43 and 40 frames were randomly extracted. These formed our hand image data set,  $H_A$ ,  $H_B$  and  $H_C$ , totalling 123 images of  $352 \times 288$  resolution. The main feature of this data set is that it contains images of a specific object on different background and with different lighting conditions. The hand boundary in all images were manually outlined and represented by a polygon with 39 landmarks. Due to the large shape variation in these images, a two-level Gaussian Pyramid is built to perform the multi-resolution scheme in both original and improved ASM methods.

The percentage of profiles from set  $H_A+H_B+H_C$  that are included by the main three classes are presented in Fig. 4. Because of high noise level and complex intensity variation, the number of components needed to model the profiles are greater than 3 for almost all landmarks. Four clear troughs can be observed for both levels at landmarks 8-11, 14-16, 22-26 and 30-33, which are located between each two fingers on the hand boundary. This indicates that landmarks at these locations have more complex intensity variation which is caused by finger movements. Obviously, a single Gaussian is not sufficient to represent such an intensity variation and may produce less appropriate search results. Furthermore, landmarks 4-6, 13, 19-21, 27-29 and 36-37 lie at five finger tips respectively and present less intensity variation. As a result, fewer Gaussian components are used to represent the distribution of intensity profiles at these landmarks.

Leave-one-image-out experiments were performed on the three subsets respectively, and subsequently on four combinations of the subsets:  $H_A+H_B$ ,  $H_A+H_C$ ,  $H_B+H_C$  and  $H_A+H_B+H_C$ , using both original and improved ASM methods. Example results are given in Fig. 5. The distributions of RMSD are presented in Fig. 6. Although to a lesser extent than the synthetic data, this shows an improved performance for the described ASM approach using grey-level profile mixture modelling.



**Fig. 4.** Percentage of profiles at each landmark that belong to the main three Gaussians. Two Gaussian Pyramid levels: (a) first level, (b) second level.

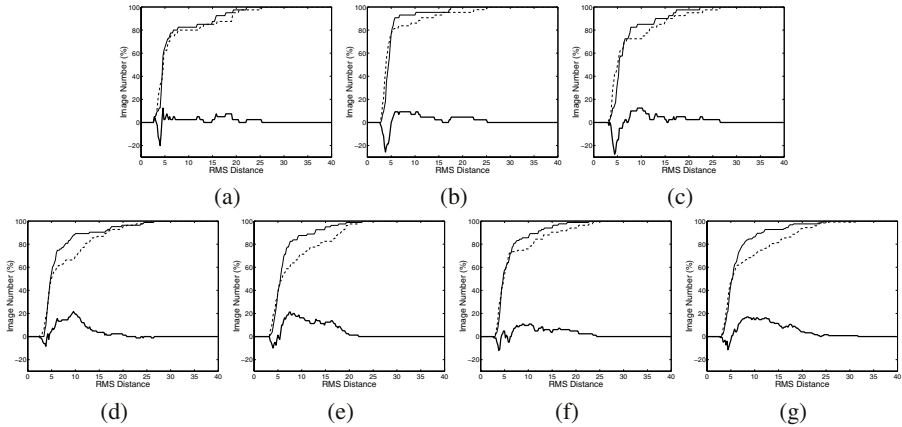


**Fig. 5.** Example results for the segmentation of the hand images in set  $H_{ABC}$  (white solid lines): (a) results of original ASM; (b) results of mixture-model ASM on the same images.

Table 1 gives a comparison of time used to build the models and perform boundary search using both methods. The mixture-model ASM method spends more time on modelling because of the iterative optimization process in the EM algorithm, which is time-consuming specially for large data sets. However, the mixture-model ASM method uses less time at the search stage (see Table 1), which implies a faster convergence speed than the original ASM.

## 4 Discussion and Conclusions

We have investigated the use of Gaussian mixture models to represent the distribution of intensity profiles and, by applying this technique in ASM modelling and search, we improved the performance of ASM method when applied to segmentation of images with complex intensity variation. Generally, the ASM method using the Gaussian mixture model framework produces faster convergence and higher robustness than the original



**Fig. 6.** Results on hand video images. The distribution of RMSD, between the ground-truth and original ASM (dotted lines), the ground-truth and improved ASM (thin solid lines). The difference of RMSD between two methods are shown as thick solid lines. Here (a) Set  $H_A$ , (b) Set  $H_B$ , (c) Set  $H_C$ , (d) Set  $H_{AB}$ , (e) Set  $H_{AC}$ , (f) Set  $H_{BC}$  and (g) Set  $H_{ABC}$ .

**Table 1.** A comparison of average model training and search time.

Data	Average training time (sec.)		Average search time (sec.)	
	Original ASM	Improved ASM	Original ASM	Improved ASM
$S_A$	0.406	66.688	0.537	0.299
$S_B$	0.375	85.609	0.664	0.360
$S$	0.718	224.563	1.634	0.420
$H_A$	0.234	71.360	0.722	0.483
$H_B$	0.235	78.859	0.491	0.380
$H_C$	0.235	62.594	0.665	0.456
$H_{AB}$	0.532	232.687	0.969	0.867
$H_{AC}$	0.515	192.969	0.976	0.866
$H_{BC}$	0.547	210.406	0.707	0.747
$H_{ABC}$	0.766	327.375	1.165	0.839

ASM. This improvement is based on a more accurate representation of the intensity variation in the images.

There is a significant difference in segmentation accuracy in moving from synthetic to real images. A main reason for this is that, in synthetic images, the landmarks are perfectly located on the real shape boundary hence the shape model and profile model can precisely represent the distributions, while manual annotation of the real images causes less accurate boundary position and lower landmark correspondence. Furthermore, the shape variation in hand images is much larger and more complex than in the synthetic data sets.

This methodology can be applied to other choices of features used to determine the landmarks positions during ASM search, such as texture information mentioned

in [7]. Since complicated intensity variation can be modelled using mixture modelling, the improved ASM method can be applied to those segmentation tasks with diverse intensity variation, such as registration in multi-modality medical imaging, and tracking objects in videos with variable object or background intensities.

Cootes et al. [2] have presented their work on improving ASM by using Gaussians mixture model to represent the *shape* variation. Since we concentrate on the use of Gaussian mixture model for profile *intensity* variation, theoretically, a combination of both methods will largely improve the robustness and efficiency of the ASM approach. Another significant improvement to ASM was proposed by van Ginneken et al. [7], which used optimal image features and  $k$ NN classifiers for ASM search. It is not unreasonable to assume that this variation can produce better results than the original ASM on the data sets we used to evaluate our method. Further work will be undertaken to make a comprehensive evaluation of ASM using mixture models for grey-level profiles compared to other variations on the basic ASM approach.

## Acknowledgements

We gratefully acknowledge partial support from Prostate Research Campaign UK.

## References

1. T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6)(2):355–366, 1994.
2. T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. *Proceedings of British Machine Vision Conference*, 1(3):110–119, 1997.
3. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1)(1):38–59, 1995.
4. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39:1–38, 1977.
5. M. Rogers and J. Graham. Robust active shape model search. *Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York*, 2353(4):517–530, 2002.
6. C. J. Twining and C. J. Taylor. Kernel principal component analysis and the construction of non-linear active shape models. *Proceedings of British Machine Vision Conference*, 1:23–32, 2001.
7. B. van Ginneken, A. F. Frangi, J. J. Staal, B. M. ter H. Romeny, and M. A. Viergever. Active shape model segmentation with optimal features. *IEEE. Transactions on Medical Imaging*, 21(8):924–933, 2002.

# Human Figure Segmentation Using Independent Component Analysis

Grégory Rogez, Carlos Orrite-Uruñuela, and Jesús Martínez-del-Rincón

Aragon Institute for Engineering Research  
University of Zaragoza, María de Luna 1, 50018 Zaragoza, Spain  
{grogez, corrite, jesmar}@unizar.es

**Abstract.** In this paper, we present a Statistical Shape Model for Human Figure Segmentation in gait sequences. Point Distribution Models (PDM) generally use Principal Component analysis (PCA) to describe the main directions of variation in the training set. However, PCA assumes a number of restrictions on the data that do not always hold. In this work, we explore the potential of Independent Component Analysis (ICA) as an alternative shape decomposition to the PDM-based Human Figure Segmentation. The shape model obtained enables accurate estimation of human figures despite segmentation errors in the input silhouettes and has really good convergence qualities.

## 1 Introduction

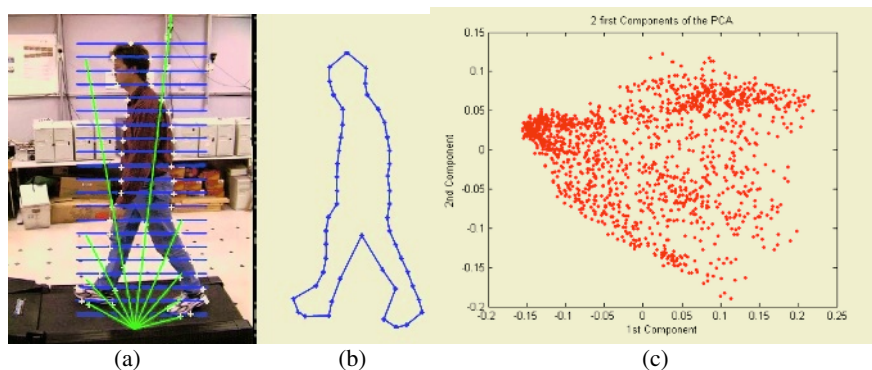
Many works have attempted to accurately estimate the shape of the human body along a video sequence using 2D or 3D models of the object contour (e.g., see [1, 2]) and principal component analysis (PCA) is largely used in this purpose. In a previous work [3] we proposed a statistical model for detection and tracking of human silhouette, based on PCA, and the corresponding 3D skeletal structure.

Following this approach, a shape model is generated from a training set (see Fig.1.) extracting the mean shape and the variation modes using PCA. Then the model is fitted to the silhouette extracted from the image by background subtraction and estimation is made of the human posture according to the contour obtained. The determination of the contour is a key factor for a good posture valuation. The more precise the segmentation is, the more accurate the estimate.

The main problem is that the PCA assumes a Gaussian distribution of the input data. This supposition fails because of the non-gaussianity of the feature space, as Figure 1(c) illustrates in the Human figure case. This non-gaussianity of the landmarks distribution is mainly caused by the non-linearity of the shape variation. This non-linearity is the result of natural curvature of the model: key points of the model move in a non-linear fashion within the image frame.

This may lead to a wrong description of the dataset and cause bad effects on the model that can model implausible shapes or cannot generate shapes that are desired. In the Figure 6, we show how a bad detection of the body, e.g. a silhouette badly segmented gives an unsatisfactory estimation of the contour when a good shape-model should help to find a correct and plausible shape that fits the blob.





**Fig. 1.** (a) Contour extraction: the positions of 49 points of each shape are considered. The images from the “CMU Motion of Body Database” [4] have been used. On each picture, the 2D coordinates of the 49 landmarks have been taken manually or semi-automatically and stored in shape vectors. (b) shows a resulting contour. We processed the sequences of 15 people (2 walking cycles) in a lateral view. After aligning and scaling the 2000 shapes, we generate our ASM model by PCA. The data projected onto the 2 first modes is represented on (c).

This drives us to search for a new approach to generate our model. The Independent Component Analysis [5] has produced some encouraging results in the Biomedical Image Processing area [6]. ICA differs from PCA in that it seeks such directions in feature space that are most independent from each other instead of directions that represent data best in a least squares sense.

There are two main problems to be considered when using ICA as has been noted in [7]. First the reliability of the estimated independent components is unknown: we ignore which of the components are to be considered seriously. The further problem is that most algorithms have random elements and every run gives different results.

The goal of this work is to generate a reliable statistical shape model for human figure segmentation. This is achieved by using ICA to model the shape variations. In this paper we demonstrate the potential of ICA in non-linear shape modeling applying it to the human figure case. Section 2 describes shape modeling with ICA. In section 3, we apply the validation method to obtain a reliable model. Then we give some result in section 4, followed by some discussion in the conclusions section.

## 2 ICA Modeling of the Human Figure

ICA, also known as Blind Source Separation, is originally used for finding source signals from mixtures of unknown signals, without any knowledge other than the observation. It can be used too for feature extraction [8]. If we consider a human shape as a mixture of source signals (a source shapes), we can illustrate it as follows:

$$dX = AS \quad , \quad (1)$$

where  $A$  is the matrix of mixing parameters,  $S$  the source shapes and  $dX$  is the matrix of the training set, that will be defined as the matrix of the variations of the  $n$  training shape-vectors with respect to the mean shape:

$$dX_i = X_i - \overline{X}, \quad i = 1 \dots n, \tag{2}$$

where the  $X_i$  are the training shape-vectors and  $\overline{X}$  the mean shape. To prevent the data from overlearning we pre-processed it by PCA [5]. The goal of the Blind Source Separation is to estimate the de-mixing matrix  $W$  that will give an estimation of the original source shapes:

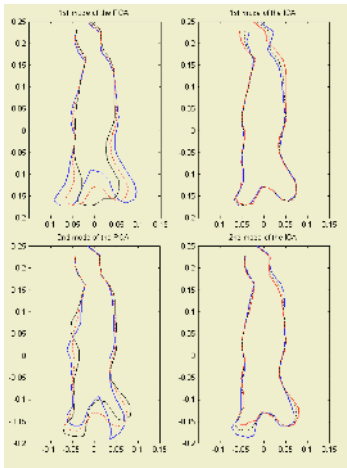
$$\hat{S} = \hat{W} dX. \tag{3}$$

The de-mixing matrix can be found using different methods. In this work we used the FastICA algorithm developed by Hiväriinen and Oja [9]. As in PCA case, the ICA model is constructed by combining the mean shape and the variation of each mode. The linear generative model is formulated as follows:

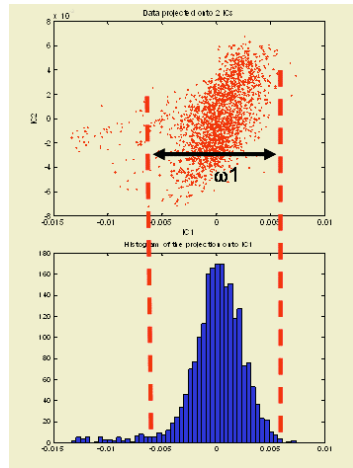
$$X \approx \overline{X} + \hat{S} b, \tag{4}$$

where  $b$  is the weighted coefficient vector.

If we vary the corresponding weight factor of an Independent Component, we can observe a variation with respect to the mean shape with certain amplitude (See Fig.2). To quantify the amplitude of the shape variation, we used a method given in [6]. We project all the shapes onto each IC and compute a histogram which width  $\omega$  is considered as a measure of variation. To discard outliers and eliminate part of the noise, the width  $\omega$  of the histogram is calculated as follow: parting from the median value, the “surface of interest” of the histogram is determined by summing the values until a percentage of the total surface is reached (Fig.3.)



**Fig. 2.** Modes of PCA (left) and ICA (right) models.



**Fig. 3.** Determination of  $\omega$  with 95% of the surface considered.

Figure 2 shows two ICA derived shape variation modes. For comparison, the two first PCA derived shape variation modes are also shown. The two models have been generated with the same data. Basically, we can observe that ICA modes variations

are quite localized along the shape. For each mode, a few part of the shape varies whereas the remainder part is unaffected. On the contrary, PCA modes present a global shape variation distributed over the entire contour.

Like all the ICA algorithms, FastICA is stochastic, i.e. the result may be different in different runs of the algorithm. Thus, the result obtained after a single run of the FastICA algorithm cannot be trusted, and the reliability of the components has to be analyzed.

### 3 Validation of the ICA via Clustering

The method is based on estimating a large number of candidate independent components by running FastICA many times, and clustering the components obtained in the signal space. Each estimated independent component is one point in the signal space. We will adapt the validation of the independent components proposed in [9] to our problem. First, the FastICA algorithm is run  $M$  times. The estimates of demixing matrices from each run  $i = 1, 2, \dots, M$  are collected into a single matrix:

$$\hat{W} = [ \hat{W}_1^T \hat{W}_2^T \dots \hat{W}_M^T ]. \quad (5)$$

A good measure of the similarity between the estimated independent components is the absolute value of their mutual correlation coefficients  $r_{ij}$  elements of the matrix:

$$R = \hat{W} C \hat{W}^T. \quad (6)$$

where  $C$  is the covariance matrix of the original data  $dX$ .

Therefore, we need to transform the similarity matrix into a dissimilarity matrix with elements  $d_{ij}$ . A classic way to make this transformation is given by [10]:

$$d_{ij} = 1 - |r_{ij}|. \quad (7)$$

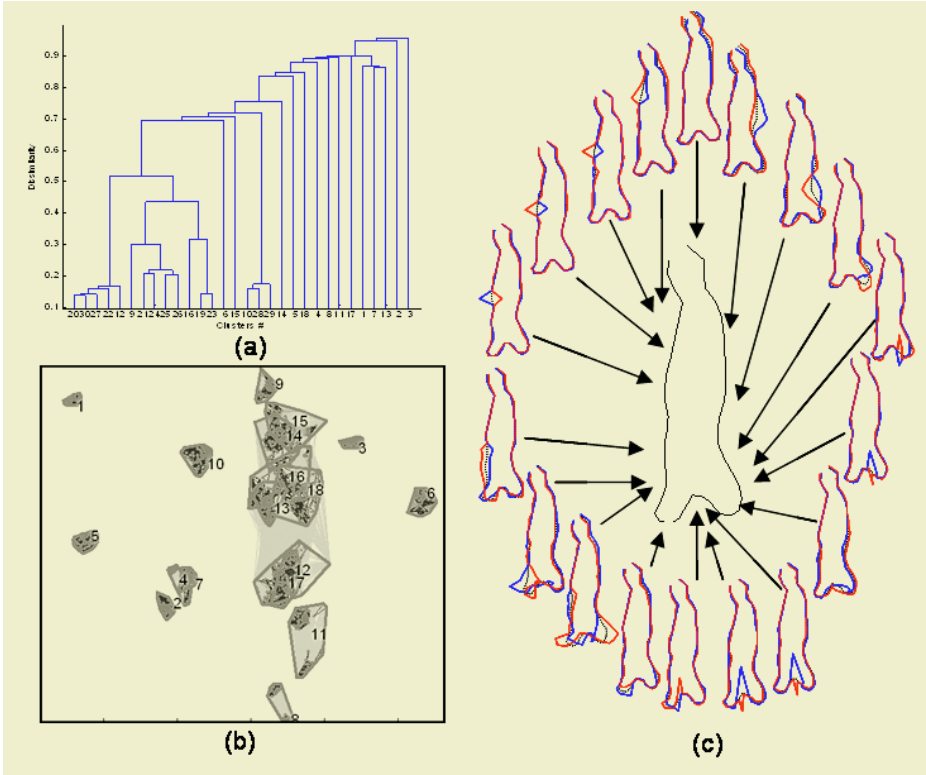
Using the dissimilarity as a measure of distance, we decompose the data into a several levels of nested partitioning (tree of clusters), called dendrogram. The points are successively joined into clusters when moving upwards in the dendrogram. A clustering of the data is obtained by cutting the dendrogram at the desired level. Then each connected component forms a cluster (See Fig.4a.).

A representative point is then computed for each cluster: we calculate the similarity intra-cluster and consider the centre of the cluster the point with the maximum sum of similarities to other points in the cluster.

To better visualize the result, we apply the Linear Discriminant Analysis for data visualization [11]. LD1 and LD2 are the first two linear discriminants that map the samples with known class from the  $n$ -dimensional space to the plane, in such a way that the ratio of the between-group variance and the within-group variance is maximized. The clusters and their interrelations are visualized in Figure 4b. We can note that there are clusters that seem to be more compact and interesting than others.

Following this methodology we have found a reliable linear non-orthogonal coordinate system. We can observe how each mode is localized along the shape and

models a particular part of the human figure. For example, some modes are associated to the movement of the hands while another models the movement of the head. Most of the ICs are associated to the legs movement. It's understandable since the principal variation that characterizes the evolution of a walking person figure is the movement of the legs. A sorting based on the position of the components along the shape is done since there is no natural sorting criterion for the components in ICA (See Fig. 4c.).

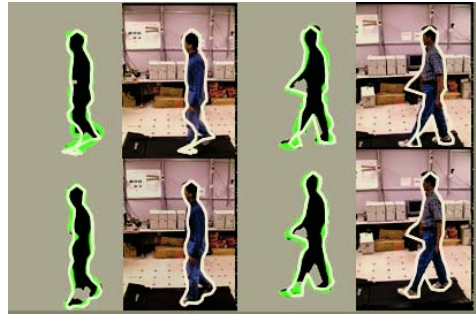
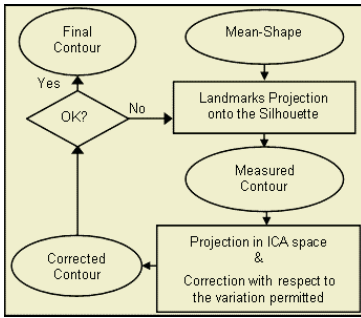


**Fig. 4.** (a) Dendrogram. Cutting it at the level dissimilarity = 0.1 gives 30 clusters when cutting it at 0.4 gives 18 clusters. (b) Similarity graph of the estimates. Clusters are indicated by convex hulls. Lines connect estimates whose similarity is larger than a threshold, the darker the line the stronger the similarity. (c) The 18 Variation Modes obtained ordered along the shape.

## 4 Experimental Results

### 4.1 Results Improvement in Cases of Bad Detection

Our ICA-based model is now applied on images where our previous PCA-based model [3] fails because of the bad detection. It is iteratively deformed to fit to the blob (the silhouette) extracted from these images (See Figure 5). Some results are shown in Figure 6. The implausible shapes generated by PCA are corrected with ICA.



**Fig. 5.** Iterative Algorithm of the Contour Segmentation.

**Fig. 6.** Segmentation using PCA (up) and ICA (down) based models.

### 4.2 Numerical Results

The model is now applied on a set of images of walking people from the MoBo database [4] that we previously processed manually to determine the contour of the person. We will measure how close from this “good contour” is the one estimated with our model. In that way we define the metrics used for the evaluation of the performances. Two distances between shapes are considered.

Suppose  $S_i$  and  $S_j$  are 2 shape vectors  $(x_{i,1} \dots x_{i,n}, y_{i,1} \dots y_{i,n})$  and  $(x_{j,1} \dots x_{j,n}, y_{j,1} \dots y_{j,n})$ , firstly, a Euclidean distance  $D_{ij}$  between these two shapes is given by:

$$D_{ij} = \sqrt{\sum_{k=1}^n ((x_{i,k} - x_{j,k})^2 + (y_{i,k} - y_{j,k})^2)}. \tag{8}$$

We also define a “Point to Curve” distance  $D_{ij}$  between the landmarks of  $S_i$  and the curve formed by the segments interpolated between the landmarks of  $S_j$ . Since  $D_{ij}$  and  $D_{ji}$  can have different values, we define this distance  $D$  as the mean value:

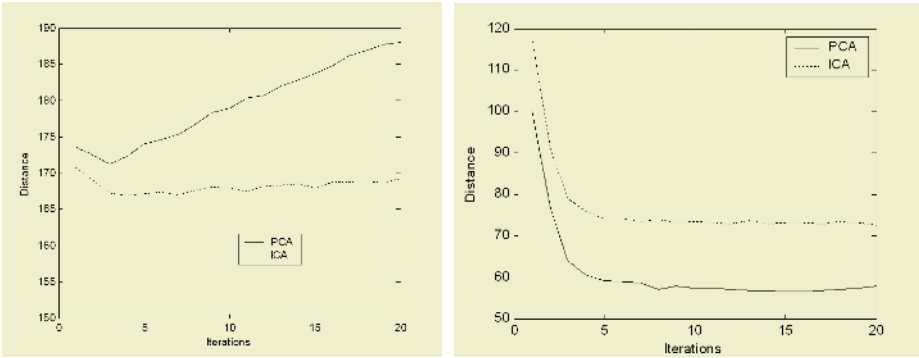
$$D = \frac{(D_{ij} + D_{ji})}{2} = \frac{1}{2} \sum_{k=1}^n (d_{i,j,k} + d_{j,i,k}), \tag{9}$$

where

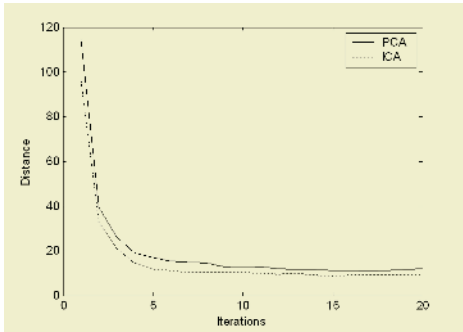
$$d_{i,j,k} = \min(\sqrt{(x_{i,k} - x_M)^2 + (y_{i,k} - y_M)^2}, (x_M, y_M) \in \text{curv}(S_j)). \tag{10}$$

The idea of this new metric is to get a null distance between two contours that differ only by a displacement of some landmarks along the shape and allow a better measure of convergence. It is to note that using this distance makes sense only if the Euclidean distance has a reasonable value: for example a shape vector containing all its components equal to one component of another contour would have a null distance with it though the two shapes are totally different.

We apply now our model on a set of 450 images (30 pictures of 15 persons) and consider 20 iterations of the algorithm for each image. The Euclidean distances of the current corrected shape with the “good” shape, and with the measured shape (determined on the silhouette blob) are calculated at each iteration. For each one of the distances calculated, a mean value is represented. In order to evaluate the results ob-



**Fig. 7.** Results obtained by the PDM based Human Figure Segmentation using PCA or ICA: Euclidean distances between corrected contour and “good contour” (*left*), and between corrected contour and measured contour (*right*) are given.



**Fig. 8.** Results obtained by the Human Figure Segmentation using PCA or ICA: “point to curve” distance between the corrected contour and the previous one.

tained with ICA, we compare them to the ones obtained with PCA using the same number of components. Figure 7 shows the results we obtained.

We can note how the distance to the “good” contour reaches its lowest value after 3 iterations with both methods and then starts to increase in the PCA case when it stays quite stable with ICA. We also can observe how the distance to the measured contour converges in both cases but with a lower value of convergence with PCA than with ICA. This can be explained by the fact that the PCA model fits exactly the blob and its eventual defects while the ICA model corrects them.

The distance between the current corrected shape and the previous one is now calculated at each iteration to evaluate the convergence of the results (See Figure 8). In both cases there is convergence, but the ICA method converges faster than the PCA one. It’s mainly due to the fact that ICA method has local variations whereas with PCA the variation is global: for each iteration, the ICA model changes local parts of the shape while the PCA one moves quite all the landmarks.

## 5 Conclusions

This work shows the potential of the Independent Component Analysis as an analysis tool for extracting local shape variations. Indeed the ICA gives a representation of the training dataset, which consists of vectors that describe local deformations, whereas the vectors obtained by Principal Component Analysis describe global deformations.

The first evaluation of the Human Figure Segmentation using ICA produces some encouraging results. Our shape model enables accurate estimation of human figure despite segmentation errors in the input silhouettes and has really good convergence qualities: compared with the PCA method, the convergence is obtained faster.

We propose a new metric to measure this convergence. In a future work, we could analyze the possibility of using this convergence in the human detection task, deciding if the input silhouette is human or not. A more complete study would have to be done to test and select the different settings.

## Acknowledgments

G. Rogez is supported by a FPU grant AP2003-2257 and J. Martínez del Rincón is supported by a FPI grant BES-2004-3741 both from the Spanish Ministry of Education. This work is also supported by a grant TIC2003-08382-C05-05 from the Spanish Ministry of Sciences and Technology.

## References

1. A. Baumberg and D. Hogg. Learning deformable models for tracking the human body, in M. Shah and R. Jain (Ed.), *Motion-Based Recognition*, 3 (Dordrecht: Kluwer, 1997) 39-60.
2. A. Blake and M. Isard. *Active Contours*, (Springer-Verlag, 1998).
3. C. Orrite-Uruñuela, J. Martínez del Rincón, J.E. Herrero Jaraba, G. Rogez: 2D Silhouette and 3D Skeletal Models for Human Detection and Tracking. *ICPR* (4) 2004: 244-247
4. The CMU Motion of Body (MoBo) Database, <http://www.hid.ri.cmu.edu>
5. A. Hyvärinen, J. Karhunen and E. Oja, *Independent Component Analysis*. (Wiley Interscience, 2001)
6. Üzümcü, M., Frangi, A.F., Reiber, J.H., Lelieveldt, B.P. Independent Component Analysis in Statistical Shape Models. In Sonka, M., Fitzpatrick, J.M., eds.: *Proc. of SPIE*. Volume 5032. (2003) 375-383
7. J. Himberg, A. Hyvärinen and F. Esposito, Validating the independent components of neuroimaging time-series via clustering and visualization. *Neuroimage*, 22:3, pp.1214-1222, (2004).
8. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J. Face Recognition by Independent Component Analysis. *IEEE Trans. on Neural Networks* 13 (2002) 1450-1464
9. A. Hyvärinen and E. Oja. A Fast Fixed-Point Algorithm for Independent Component Analysis, *Neural Computation*, 9(7), pp. 1483-1492, 1997
10. B. Everitt, *Cluster Analysis*. Edward Arnold, London, third Edition (1993).
11. Jaakko Peltonen and Samuel Kaski. Discriminative Components of Data. *IEEE Transactions on Neural Networks*, accepted for publication.

# Adaptive Window Growing Technique for Efficient Image Matching

Bogusław Cyganek

AGH – University of Science and Technology  
Al. Mickiewicza 30, 30-059 Kraków, Poland

**Abstract.** The paper presents a new approach to image matching based on the developed adaptive window growing algorithm. This integer-only algorithm operates on monochrome images transformed into the Census nonparametric representation. It effectively computes the entropy of the local areas and adjusts their size if the entropy is not sufficient. This way the method allows for avoidance of featureless areas that cannot be reliably matched, at the same time maintaining the matching window as small as possible. The special stress has been also laid on efficient implementation that can fit the custom hardware architectures. Therefore the presented algorithm requires only an integer arithmetic. Many experiments with the presented technique applied to the stereovision matching showed its robustness and competing execution times.

## 1 Introduction

Block matching plays an important role in the computer vision. It is widely used method for visual tracking, stereovision, video compression, etc. An inherent problem associated with every block matching is choice of a shape and size of the matching region, as well as the range of the search throughout the images. Unfortunately these choices are not unique since they depend greatly on application and image contents. Sometimes the search range can be preset based on application, e.g. for maximum disparity in the stereo-matching the statistical analysis based on variograms can be used [3] or in motion analysis the search range can be restricted to few pixels in every direction. However the shape and size of a matching window is usually unknown beforehand. Therefore the simplest method is to use a rectangular window with fixed size. However, regardless of the employed matching measure, such an approach leads to matching errors in a form of either false matches or excessive blurring. At the one hand, the window size must be large enough to convey sufficient information for a reliable match, but at the other hand the window should be as small as possible to comprise only pixels with features of the same object and to avoid blurring of the output disparity maps. This is why the adaptive window concept has been developed.

The adaptive window technique proposed by Lotti [8] for matching aerial images relies on sophisticated window growing that is limited by edges and statistical contents of the matching regions. However, this technique is quite time consuming.

One of the very original solutions with a statistical model was proposed by Kanade and Okutomi [7]. The appropriate window is selected by evaluating the variations in



intensity and disparity. The idea is that at discontinuities the intensity and disparity variations are larger, unlike at the positions of surfaces.

In another class of methods, known as the multiple windowing [5], a small number of different windows are used for matching, and the one with the best cost is retained. Usually window size is fixed, but its shape is changed. However, the methods of this class show up some problems in low texture regions.

Just recently Veksler proposed the new method of variable windows for stereo correspondence [10]. This method is based on the concept of image integrity.

Unfortunately, most of the aforementioned methods are rather time consuming and usually require advanced numerical computations. As a result they are not very well fitted for the real time and custom logic implementations.

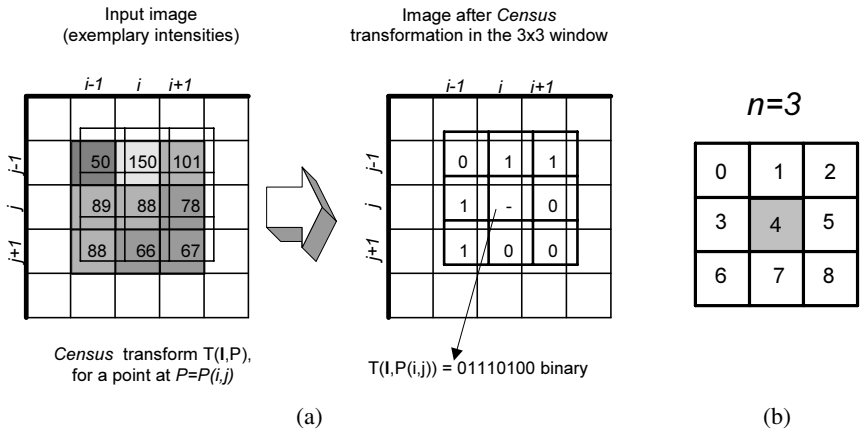
The method proposed in this paper assumes the nonparametric image transformation *before* the window adapting mechanism is used. This is a novel approach to the matching problem, not explored by the other authors. As a nonparametric transformation we use the *Census* and *Rank* methods, first proposed by Zabih and Woodfill [11], then employed with success to the real time stereo system [1]. The Adaptive Window Growing technique (AWG) proposed in this paper adapts size of the matching window as to maximize an amount of conveyed information in the information-theoretic sense. This is done thanks to the observation of entropy increase, up to a certain value, with an increase of zero-value-bits in the *Census* representation of matching windows. Such an approach does require only simple *integer arithmetic*. Therefore the method is appropriate for hardware implementation and real time processing.

## 2 The Adaptive Window Growing Technique in Census Domain

The *Rank* and *Census* transforms were proposed by Zabih and Woodfill [11] for computation of correspondences by means of the local nonparametric transformation applied to the images before matching process. Both transformations start in the image intensity signals domain and are computed in a certain compact region around a central pixel. Size and shape of this region can be set arbitrarily, usually it is a square. Such square regions are also assumed in this paper, although this assumption can be relaxed.

For a given central pixel and its closest neighborhood the *Rank* transform is defined as the number of pixels in that region for which the intensity signal is greater or equal than intensity of the central pixel. For the same setup the *Census* transform returns an ordered stream of bits where a bit at a given position is set if and only if the central and corresponding pixels hold the same relation, i.e. an intensity value at that pixel is greater or equal to the one at the central pixel. Fig. 1a explains the ideas behind the *Census* transformations. Fig. 1b depicts assumed pixel orders for computing *Census* values for the  $3 \times 3$  square neighborhood of pixels. An interesting observation for *Census* is that a value of the central pixel is taken only as a reference and does not go into the output bit stream. Therefore for  $3 \times 3$  and  $5 \times 5$  regions we obtain computer efficient representations of eight and twenty four bits (i.e. one and three bytes), respectively.

The *Census* transform maps the local pixel neighbourhoods, located around a certain central pixel  $P$ , to a bit string. In this series each bit conveys a 0/1 information,



**Fig. 1.** The *Census* transformation for a pixel at position  $(i,j)$  (a), the assumed pixel numbering in the square  $3 \times 3$  window  $W$  for computation of the *Census* transformation (b)

saying whether a given pixel is less or not from the central pixel. The *Census* transform for a pixel  $P$  in the image  $I$  is given as follows [11]:

$$T[I, P] = \bigotimes_{P \in W(P, \beta)} \xi(I, P, P') \tag{1}$$

where  $I$  denotes intensity,  $P$  is a central pixel,  $\bigotimes$  denotes concatenation,  $W(P, \beta)$  is a local pixel neighbourhoods around pixel  $P$  with radius  $\beta$ ,  $P'$  denotes pixels belonging to  $W$ , and  $\xi$  is given by the following formula:

$$\xi(I, P, P') = \begin{cases} 1 & \text{if } I(P) \leq I(P') \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

To find correspondence of images we first apply (1) to the images and then usually compute the Hamming distance between bit strings, although some other measures are also possible [4]. The other question is choice of the window  $W$  [4].

### 2.1 Adaptive Window Growing

In this section we explain the AWG method. The basic idea of this technique relies on observation of entropy increase in a certain neighborhood of pixels with increase of '0' value bits in the *Census* representation of this neighborhood.

From the definition (2) and from the mutual relation between the central point 'K' and all its closest neighbors 'a' – 'h' in Fig. 2 we see that if only an intensity value of the point 'K' is different from all values of the points 'a' – 'h' we have fulfilled the condition for maximum entropy in this point arrangement and at the same time the maximum number of '0's is achieved. This happens because if only the pixels differ then a bit with value '0' will be assigned for this relation in accordance with (2). This bit will be either in the *Census* bit stream representation for the pixel 'K' or in its

corresponding pixel. Nevertheless, this ‘0’ will count for the whole 3×3 neighborhood. There exists also another type of the mutual pixel relation – the relation among three consecutive pixels, such as ‘a’, ‘b’, and ‘d’ in Fig. 2. For each pair of abutted pixels from such a triple, a bit with value ‘0’ can be assigned if and only if any pair of intensity values is different. In such a case, for each pair of pixels from the triple, one *Census* representation obtains ‘1’ at the corresponding bit position, while the same bit position – but in the complementary pixel from this pair – holds ‘0’. This is clear from (2) since there is an exclusive relation. Thus, if only the abutted pixels have different values, at the pertaining bit position the only possibility is: bit ‘1’ in the first *Census* pixel of a pair and ‘0’ for the second one.

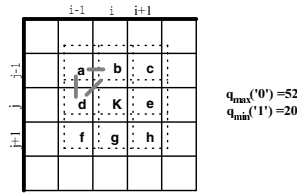


Fig. 2. Mutual relations of pixel values for the *Census* transformation ( $q$  is a number of bits)

The above analysis leads to the following rules for obtaining the maximum number of ‘0s’ – and in consequence increase of entropy – in a *Census* representation of any 3×3 neighborhood (Fig. 2):

1. A value of the central pixel  $K$  is irrelevant provided that it is different than any other pixel from its closest neighborhood (i.e. from the all pixels  $a, b, c, d, e, f, g,$  and  $h$  in Fig. 2):

$$\forall p \in N - \{K\}: I(p) \neq I(K), \tag{3}$$

where  $I(p)$  is an intensity value of a pixel at index  $p$  (see Fig. 1b) from the 3×3 neighborhood  $N$ .

2. Any corner-triple of pixels (such as e.g.  $a, b, d$  in Fig. 2) must be different, i.e.:

$$I(a) \neq I(b) \neq I(c). \tag{4}$$

3. All other pixels bordering with pixels from the neighborhood  $N$  must have their intensity values less than their direct pixels-neighbors from  $N$ . For example, for the pixel  $b$  in Fig. 2 these would be pixels at indexes:  $(j-2,i-1), (j-2,i),$  and  $(j-2,i+1)$ .

The lower bound for data entropy (LBE) in any 3×3 *Census* neighbourhood that preserves the conditions 1-3 is obtained by taking only four different values and can be easily found from [6] to be:  $LBE = -4/9 \times \log 4/9 - 2 \times 2/9 \times \log 2/9 - 1/9 \times \log 1/9 \approx 0.55$ . The upper bound for the entropy in any 3×3 neighbourhood is  $MBE = \log(9) \approx 0.95$  (i.e. nine different values). Thus, any 3×3 neighbourhood of pixels that preserve conditions 1-3 guarantee almost 58% of maximum possible entropy in this neighbourhood.

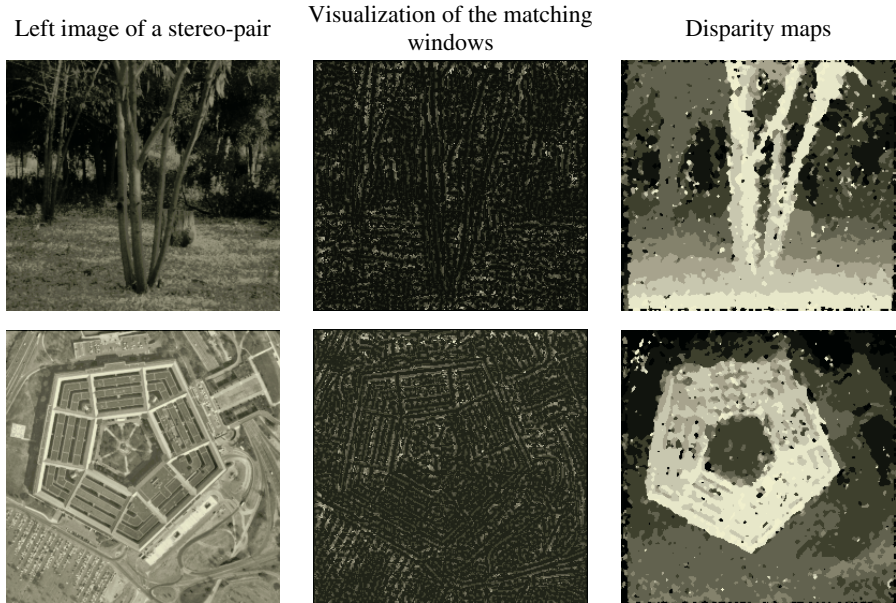
The upper bound for any *Census* in a square neighbourhood 3×3 is  $q_{max}(3, '0') = 52$  and follows easily the conditions 1-3 and Fig. 2. Concluding we can state that the set of conditions 1-3 guarantying the maximum number of ‘0’s in any 3×3 *Census*



From the presented experiments it is evident that the AWG technique is very efficient for images with large blobs of the same intensity values. The non uniformity of the adaptive windows in the central part of image from Fig. 3c comes from the settings of a large upper size  $n_{max}=33$  of windows. Therefore when the window comes to its upper allowable size the other features got caught (the walls of the corridor) and in result the computed output size of that window is smaller than  $n_{max}$ .

### 3.1 Adaptive Window Technique Applied to the Stereovision

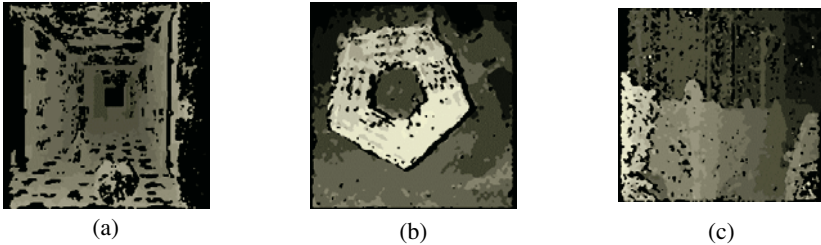
The presented adaptive matching technique was employed in stereovision matching to show its quality and potential applications. The results are quite promising and very competing to the other matching methods, even those with very complex – and time consuming – comparison measures. The presented stereovision method does not require any floating or even fixed point arithmetic – only integer arithmetic is used. Therefore the method is appropriate for custom hardware implementation if real time applications are taken into consideration.



**Fig. 4.** Adaptive Window Growing applied in the stereo matching with *Census* measure. From the top: image from the Tsukuba University, Map provided by Scharstein and Szeliski [9], Trees SRI and Pentagon from the CIL CMU. From left to right: a left image from each pair, visualization of sizes of the matching windows determined by the AWG algorithm (brighter places denote larger matching windows), disparity maps (brighter places denote closer objects)

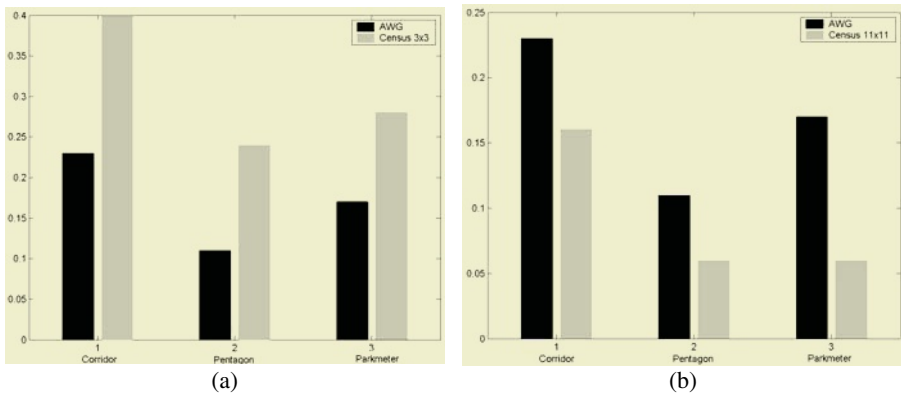
For each pixel location the size of a matching window is determined with the AWG technique defined by (5). Disparity values are found for each pixel by matching square windows of sizes appropriately set in the previous step. As a matching meas-

ure the Hamming distance is computed from the *Census* representation of each pair of candidate windows. Matching is done with a fast block matching algorithm based on the winner-update strategy with hashing table [2]. The running times in seconds on IBM with Pentium 1.5 GHz were as follows (from top of Fig. 4): 5.48, 1.52, 0.86, and 0.89 for different maximal disparities. The visible limited occlusions are due to a simple stereo algorithm without the cross-checking [12]. However, the details were preserved due to adaptively chosen matching windows.



**Fig. 5.** Cross-checked disparity maps: Corridor (a), Pentagon (b), Parkmeter (c)

The cross-checked [12] disparity maps are presented in Fig. 5. Their computation was done in a parallel fashion by use of different threads for computation of the two left-right and right-left disparity maps. Therefore the execution times are only about 20% greater than already presented. The results and visual quality of the disparity maps in Fig. 4 and Fig. 5 can be classified as very good.



**Fig. 6.** Comparison of quality of the disparity maps measured as a ratio of cross-checked rejected points to the total amount of pixels (the lower bar, the better quality): AWG matching vs. fixed 3×3 *Census* matching (a), AWG matching vs. fixed 11×11 *Census* matching (b)

The number of rejected points during the cross-checking can be used to infer a quality of the stereo matching process and for images in Fig. 5 the normalized (to the number of total pixels in the output disparity map) values of rejections are: 0.23, 0.11, and 0.17, respectively. Fig. 6 presents a qualitative comparison of the disparity maps. The quality is measured as a ratio of cross-checked rejected points to the total amount of pixels. Although this measure is not perfect it can characterize a given

stereo method from the qualitative point of view. The disparity maps obtained with the AWG (the dark bars in Fig. 6) were compared with the ones obtained from the fixed window matching (red bars):  $3 \times 3$  *Census* matching (Fig. 6a) and  $11 \times 11$  *Census* (Fig. 6b). It is evident that in the first case the AWG outperforms its counterpart.

## 4 Conclusions

The paper presents the novel Adaptive Window Growing technique that is based on the two nonparametric local transformations: *Rank* and *Census*. It has been shown that the increase of the quantity of the zero-value-bits  $q('0')$  in any *Census* neighborhoods leads to an increase of the conveyed information, conceived in the information-theoretic sense as an entropy value. The AWG method looks for a minimum size square windows that maximize  $q_{av}('0')$ , i.e. the average number of '0' bits per pixel in the *Census* representation in a given window. In effect, the subsequent matching is locally well posed and can be done much more reliably than in a case of fixed windows. The experimental results with the AWG technique applied to the stereo matching showed robustness of this technique. The main purpose of avoiding low-textured areas was achieved by means of very simple integer arithmetic. Therefore the AWG method seems to be appropriate for hardware and real-time implementations.

## References

1. Banks J., Bennamoun M., Corke P.: Non-Parametric Techniques for Fast and Robust Stereo Matching. CSIRO Manufacturing Science and Technology, Australia (1997)
2. Chen, Y-S., Hung, Y-P., Fuh, C-S.: Fast Block Matching Algorithm Based on the Winner-Update Strategy. IEEE Trans. On Image Processing, Vol. 10, No. 8, (2001) 1212-1222
3. Cyganek, B., Borgosz, J.: Maximum Disparity Threshold Estimation for Stereo Imaging Systems via Variogram Analysis. ICCS 03, Russia/Australia (2003) 591-600
4. Cyganek, B.: Comparison of Nonparametric Transformations and Bit Vector Matching for Stereo Correlation. Springer LNCS 3322 (2004) pp. 534-547
5. Fusiello, A. et.al.: Efficient stereo with multiple windowing. CVPR 858-863 (1997)
6. Haykin, S.: Neural Networks. A Comprehensive Foundation. Prentice-Hall (1999)
7. Kanade, T., Okutomi, M.: A stereo matching algorithm with an adaptive window: Theory and experiment. PAMI, 16(9) (1994) 920-932
8. Lotti, J-L., Giraudon, G.: Adaptive Window Algorithm for Aerial Image Stereo. INRIA Technical Report No 2121 (1993)
9. Scharstein, D., Szeliski, R.: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. IJCV, Vol. 47,1 No. 1-3 (2002) 7-42
10. Veksler, O.: Fast Variable Window for Stereo Correspondence using Integral Images. Computer Vision and Pattern Recognition (2003)
11. Zabih, R., Woodfill, J.: Non-Parametric Local Transforms for Computing Visual Correspondence. Proc. Third European Conf. Computer Vision (1994) 150-158
12. Zhengping, J.: On the Mutli-Scale Iconic Representation for Low-Level Computer Vision. Ph.D. Thesis. The Turing Institute and University of Strathclyde (1988) 114-118

# Managing Resolution in Digital Elevation Models Using Image Processing Techniques

Rolando Quintero, Serguei Levachkine, Miguel Torres,  
Marco Moreno, and Giovanni Guzman

Centre for Computer Research, National Polytechnical Institute, Mexico City, Mexico  
{quintero, sergei, mtorres, mmoreno, jguzmanl}@cic.ipn.mx  
<http://geo.cic.ipn.mx>  
<http://geopro.cic.ipn.mx>

**Abstract.** In this work, we propose a set of algorithms to manage the resolution of DEM for simulation processes. First, we present an application to handle the huge quantity of data contained in DEM for real-time rendering by discriminating the less significant elevation data. On the other hand, as a second step of the process, we extend the algorithm to increase the spatial resolution of DEM for cases when it is needed. Finally, we introduce a method for increasing spectral resolution of DEM by using a skeletonization process. The algorithms were developed to be used with raster data sets, although similar considerations can be taken for vector data sets.

## 1 Introduction

Digital Elevation Models (DEM) have gained popularity in applications for simulating natural disasters. Nevertheless, these applications require a huge amount of data. In many cases, the available data do not present enough quality for simulation processes. The Statistics, Geography and Informatics National Institute of Mexico (INEGI) produces DEM with 50 m of resolution [1][2], but some simulation processes require a better level of detail (1 m is the standard). In all cases, DEM are generated by means of contours. These have different representations and thresholds of separation. For instance, in the topological maps of INEGI, the contours are separated by 10 and 5 m near the coast. In Simulation processes like flooding simulations we need more detailed information (less than 1 m of resolution).

In this work, we propose a set of algorithms to manage the resolution of DEM for simulation processes. First, we present an application to handle the huge quantity of data contained in DEM for real-time rendering by discriminating the less significant elevation data, without changing the *semantics* of the raster data. However, we cannot improve the quality of the more relevant data to obtain additional information. On the other hand, as a second step of the process we extend the algorithm to increase the spatial resolution of DEM, in those cases. Finally, we introduce a method for increasing spectral resolution of DEM by using a skeletonization process. It is important to lineout that the algorithms mentioned above were developed to be used with raster data sets, although similar considerations can be taken for vector data sets.



By using these algorithms together, we can solve the problem of 3D data representation and generate virtual scenes, which are ready to navigate, either by simulations or by defined trajectories. In the next three sections, we present the algorithms mentioned above, as well as, some results. Finally, Section 5 describes our conclusions.

## 2 Real-Time Rendering by Decreasing Spatial Resolution

In this section we present an algorithm that allows creating virtual scenes according to the elevation and texture of the spatial data. Our proposal allows making rendering process in real-time; it reduces the quantity of the processed data.

The algorithm uses the elevation data stored in a matrix  $G$ , loaded from any source of elevation data, i.e. DEM files, elevation bitmaps, *etc.* A trivial algorithm to make this is the following:

```

RENDER(o)
1   for i = 1 to M-1
2   for j = 1 to N-1
3       RENDER-VERTEX(G [i, j])
4       RENDER-VERTEX(G [i+1, j])
5       RENDER-VERTEX(G [i+1, j+1])
6       RENDER-VERTEX(G [i, j+1])

```

Notice that this process could produce a huge quantity of data to process. In the tests that we made, we have used a DEM that produces a vertex grid of  $2048 \times 2048$  elements. It is more than 4 million of polygons. Applying space partitioning algorithms [6] and hide surface removal techniques [7], we must process a set of 500 thousands of polygons approximately. Then, processing such huge volume of data, it is necessary to decrease much more the number of polygons to process. This can be done by means of *Level of Detail* (LOD) algorithms.

In [3], [4] are presented some algorithms to decrease the LOD in complex scenes. These algorithms present three main drawbacks:

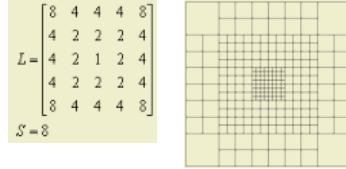
- They are complex and increase the workload of the processor. They make changes to the terrain data (spatial data), because they are focus on the final visual appearance of the scene.
- They modify the terrain data depending on the observer's viewpoint. Due to this, the spatial data analysis is not possible.
- The number of polygons rendered is variable, and then the frame per second (fps) rate is not constant during the simulation.

We have developed an algorithm to reduce the terrain LOD to speed up the data visualization facing the problems mentioned above. The goals of the algorithm are: 1) be simple, 2) not to affect the terrain data and 3) run in real-time. The algorithm must guarantee a maximum number of polygons to render (a constant fps rate).

To describe the algorithm, we must define some parameters first (their meanings are illustrated in Fig. 1).

- A matrix  $L$  of  $H \times H$  defines the discrete LOD's to use.  $H$  is an odd number greater than 1, and  $L[i,j] \neq 0$  for  $1 \leq i,j \leq H$ .
- A number  $S$  defines the optimization unit size; it means that it is necessary to optimize regions of  $S \times S$  polygons.
- A vector  $o$  represents the observer position.

Figure 1 shows the visual representation mechanism used for proposed algorithm.



**Fig. 1.** Visual representation using the parameters of the algorithm

Using the defined parameters, it is possible to outline the algorithm. The proposed algorithm is the following:

```

RENDER (o)
1   (ox, oy) ← RELATIVE-POSITION(o,G)
2   for i = -1/2H to 1/2H
3       x ← ox+S(i - 1/2)
4       for j = -1/2H to 1/2H
5           y ← oy+ S(j - 1/2)
6           RENDER-BLOCK(x,y,L[i+1/2H, j+1/2H])

RENDER-BLOCK(x, y, lod)
1   if lod > 0
2       for i = x to x+S step lod
3           for j = y to y+S step lod
4               RENDER-QUAD(i,j,lod)

RENDER-QUAD(i,j,lod)
1   RENDER-VERTEX(G[i,j])
2   RENDER-VERTEX(G[i+lod,j])
3   RENDER-VERTEX(G[i+lod,j+lod])
4   RENDER-VERTEX(G[i,j+lod])
    
```

By using the algorithm, we can easily compute the number of polygons to be processed ( $N_p$ ), given the equation (1):

$$N_p = \sum_{i=1}^H \sum_{j=1}^H \frac{S}{L[i, j]} \tag{1}$$

### 3 Bicubic Patches Based Algorithm for Increasing Spatial Resolution

In this section we present an algorithm to increase resolution of DEM for real-time simulation processes without change the semantics of the elevation data. In previous

section, we presented the algorithm to manage the huge quantity of data contained in DEM for real-time rendering. Thus, we can discriminate data from DEM. However, in the case when we need more detailed data than those that are in  $G(i,j)$  we can apply parametric patches to obtain these intermediate data. We have mentioned that elevation data can be represented as a polygon mesh. The step from polygon meshes to patch meshes is straightforward. If we consider a mesh of four-sided polygons approximating a curved surface, then a parametric patch mesh can be defined as a set of curvilinear polygons, which actually lie in the surface, and by applying parametric patches we can increase the resolution of the elevation data.

The definition given in [7] for a parametric surface (either B-spline or Bezier surfaces)  $Q(u,v)$  is in terms of two parameters,  $u$  and  $v$ , where  $0 \leq u \leq 1$  and  $0 \leq v \leq 1$ , and the function  $Q$  is a cubic polynomial. The accurate values of the coefficients in the cubic determine the curve. A special and convenient way of defining these is to use 16 three-dimensional points known as control points. The shape of the patch is fully determined by the position of these points. A bicubic surface is defined in Eqn. 2.

$$Q(u,v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} B_{i,j}(u,v) , \tag{2}$$

Where  $P_{ij}$  is an array of control points an  $B_{i,j}(u,v)$  is a bivariate basis function. We can generate  $B_{i,j}(u,v)$  in the form given in Eqn. 3.

$$B_{i,j}(u,v) = B_i(u)B_j(v) , \tag{3}$$

Where  $B_i(u)$  and  $B_j(v)$  are the univariate cubic basis function. The definition of these basis functions describes the type of surface to be generated. As a first approximation, we apply Bezier patches enable an efficient patch-splitting algorithm for rendering. But the main problem with Bezier patches is that the generated surface does not fit with the given points (control points). We cannot fit all control points with the resultant Bezier patches. Nevertheless, by applying Bezier surfaces, we can compute how the whole data set behavior affects to a single point. On the other hand, we can apply B-spline patches. Thus, we can compute the inner points between the known ones (control points), using only local information. A B-spline patch is always defined by a  $4 \times 4$  control point array. So, with this type of surface we can find the new data without affecting the behavior of the whole data set.

To integrate the increasing resolution using parametric patches, we should modify the algorithm presented in previous section. First, we should allow values less than one in matrix  $L$ . Such values mean that we expect to obtain higher resolution for the block that is being rendered. On the other hand, we need to compute the parametric curve. This curve is stored in an alternate grid called  $Q$ . The values of  $Q$  will be defined by the control points and by the transformation matrix  $B$  shown in Eqn. 4.

$$B = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \tag{4}$$

Finally, the changes in the algorithm are the following:

```

RENDER-BLOCK(x, y, lod)
1   if lod ≥ 1
2     for i = x to x+S step lod
3       for j = y to y+S step lod
4         RENDER-QUAD(i,j,lod)
5   else
6     Q ← TO-SPLINE(G,x,y,lod)
7     for i = x to x+S step lod
8       for j = y to y+S step lod
9         RENDER-QUAD-SPLINE(i,j)

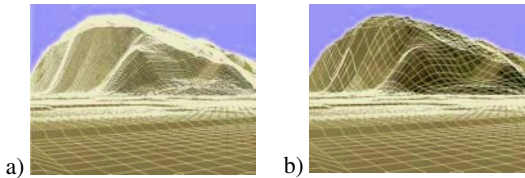
TO-SPLINE(G,x,y,lod)
1   P ← CONTROL-POINTS(x,y)
2   for u=0 to 1 step lod
3     U ← [u3 u2 u 1]
4     for v=0 to 1 step lod
5       V ← [v3 v2 v 1]
6       Q[u,v] ← U×B×P×BT×V

RENDER-QUAD-SPLINE(i,j)
1   RENDER-VERTEX(Q[i,j])
2   RENDER-VERTEX(Q[i+1,j])
3   RENDER-VERTEX(Q[i+1,j+1])
4   RENDER-VERTEX(Q[i,j+1])

```

We only present the changes for integrating B-spline surfaces. Similar considerations must be taken for applying Bezier surfaces.

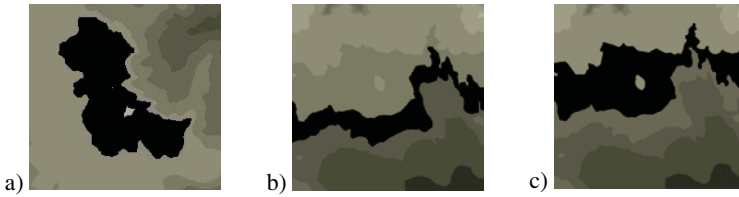
In Fig. 2 are presented some screenshots of the terrain rendering applying the algorithms outlined.



**Fig. 2.** a) Result with the trivial algorithm. b) Result with the proposed algorithm

## 4 Skeleton-Based Algorithm for Increasing Spectral Resolution

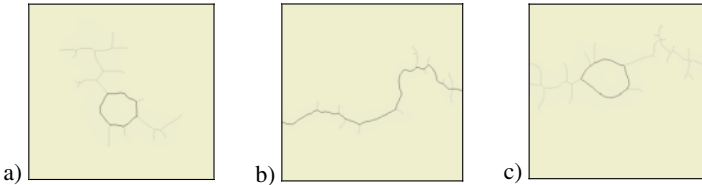
We have presented algorithms for managing spatial resolution of DEMs. Now, we will present an algorithm for increasing the spectral resolutions of elevation models. We propose the use of a skeletonization process to obtain a new contour between two known ones, i.e., to increase spectral resolution. In [5] we present the process to compute the skeleton of a binary image. Now, we define how to generate the new contour from the skeleton of the region between the two known contours. We will call this region *Equi-Height* region or EH regions. In real elevation models, many times the contours are interrupted in the edges of the image. Hence, the EH regions can be incomplete; we have identified three cases (see Fig. 3):



**Fig. 3.** Cases considered in the raster analysis

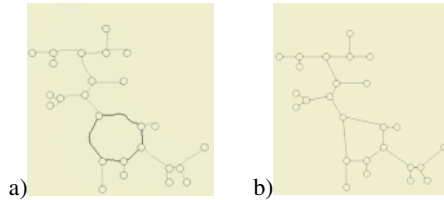
- *Case A.* The contours that define EH region are completely inside of the image, forming a blob with at least one hole (Fig. 3a)
- *Case B.* The contours begin and end outside of the image, forming a strip across the image (Fig. 3b)
- *Case C.* The same as case B, but there are holes in the strip (Fig. 3c).

In Fig. 4 we present the skeletons obtained for EH regions of Fig. 3. As we can see, each skeleton has a lot of branches that are not suitable for being part of the contour. We can discriminate the noise branches (prune the skeleton). In same Fig. 3 the noise lines are in light gray, while the contour is in black.



**Fig. 4.** The skeletons for the examples depicted in Fig. 3

To perform the skeleton pruning, we propose to generate a graph that describes the morphology of the skeleton. Thus, it is possible to find the contour of EH region by using a graph that describes the skeleton (see Fig. 5).



**Fig. 5.** Transformation of skeleton into graph

To generate the graph from the skeleton, we use on the fact that the skeleton is 8-connected. Let  $I$  be a binary image containing a skeleton and  $p_{ij}$  the value of the image matrix  $I(i,j)$ .

Let  $p$  be a pixel of the image and  $n(p)$  the number of 8-connected neighbors. Then we define  $T$  as the set of terminal pixels (Eqn. 5). Similarly, we define  $A$  as the set of

edge pixels (Eqn. 5) and  $R$  as the set of triad pixels (Eqn. 5). Also, we define  $E$  as the set of edge pixels and  $T_E$  as the set of terminal pixels that arise the edge of the image. While, the set of vertices  $V$  is given by Eqn. 6.

$$\begin{aligned} T &= \{p : n(p) = 1\}, \\ A &= \{p : n(p) = 2\}, \\ R &= \{p : n(p) \geq 3\}, \end{aligned} \tag{5}$$

$$\begin{aligned} T_E &= \{p : p \in T, p \in E\} \\ V &= \{f\} \cup T \cup R. \end{aligned} \tag{6}$$

Let  $p$  be an image pixel and  $N(p)$  the set of the 8-connected neighbors of  $p$ , then we define a branch  $s$  by Eqn. 7. Also,  $l(s)$  denotes the length of a branch  $s$ , and  $first(s)$  and  $last(s)$  defines the extreme elements of a branch  $s$ .

$$\begin{aligned} s &= \{p_1, p_2, \dots, p_n\} \ni p_i \in N(p_{i+1}), p_1 \in V, p_n \in V, \\ l(s) &= card(s) = n, \\ first(s) &= p_1, \\ last(s) &= p_n. \end{aligned} \tag{7}$$

We now define the graph representing the skeleton of the image as  $G(V, E)$  where  $E = \{s : s \text{ is a branch}\}$ . Also, we define a path  $w$  on the graph  $G$  by Eqn. 8. Moreover, we define the length of a path  $\lambda(w)$ , the set of all paths in  $G$  as  $W(G)$ , and  $\omega(G)$  is the longest path in  $G$  (Eqn. 9).

$$\begin{aligned} w &= \{s_1, s_2, \dots, s_n\} \ni s_i \in E, last(s_i) = first(s_{i+1}), i = 1, \dots, n-1, \\ w \in P_E &\Leftrightarrow first(s_i) \in T_E, last(s_n) \in T_E, \end{aligned} \tag{8}$$

$$\lambda(w) = \sum_i l(s_i) \ni s_i \in w \tag{9}$$

$$\omega(G) = w_i \ni \lambda(w_i) \geq \lambda(w_j), w_i \in W(G), w_j \in W(G), w_i \in P_E.$$

Once we have obtained the graph from skeleton, we simply take the following criteria for discriminating noise branches:

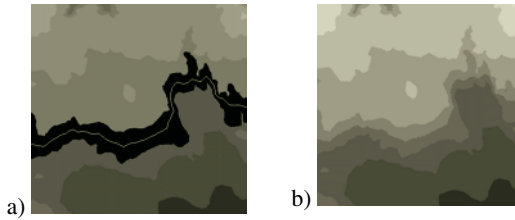
- If there are loops in graph, then all branches that are not in one loop are eliminated.
- If a branch is in more than one loop, then that branch is eliminated.
- If there are not loops in graph, then all branches outside  $\omega(G)$  are eliminated.

The next definitions are used to discriminate branches. A path  $b$  is a loop if  $last(s_n) = first(s_1)$ . So, let  $B(G)$  be the set of all loops in graph  $G$  and  $S_R(G)$  the set of redundant branches in  $G$  (Eqn. 10). Then we define the candidate contour  $C$  as is denoted in Eqn. 13.

$$\begin{aligned} S_R(G) &= \{s : s \in b_i, s \in b_j, i \neq j, b_i \in B(G), b_j \in B(G)\}, \\ C(G) &= \{s : s \in b, b \in B(G), s \notin S_R(G)\}, \end{aligned} \tag{10}$$

Finally, let  $C_N$  be the resulting contour within the skeleton, defined by Eqn. 11.

$$C_N = \begin{cases} C(G) \Leftrightarrow C \neq \phi \\ \omega(G) \Leftrightarrow C = \phi \end{cases} \tag{11}$$



**Fig. 6.** Final result. (a) Contour discovered into original DEM. (b) Modified DEM according to the source

## 5 Conclusions

In this work, we proposed a set of algorithms to manage the resolution of DEM for simulation processes. We have presented an algorithm to handle the huge quantity of data contained in DEM for real-time rendering by discriminating the less significant elevation data. The developed algorithm fully fit the stated goals. It does not overload the processor, because it is very simple. Also, the rendering algorithm guaranties a maximum number of elements to be rendered ( $N_p$ ). Additionally the algorithm does not modify the spatial data.

Similarly, we extended the algorithm to increase the spatial resolution of DEM, for cases when it is needed. An application of this algorithm in image processing is as follows: it may be used to *zoom-in* images with a non-linear re-sampling. Using parametric patches, it is possible to obtain the new values for the pixels between the known ones, containing local information (by means of B-spline patches) and global behavior (by means of Bezier patches) that improve the appearance of the enlarged image.

Finally, we have presented an algorithm for increasing spectral resolution in DEM. The algorithm is based on the 8-connected skeleton of polygons composed of the contour lines; and to prune this skeleton by transforming it into a graph. The algorithm is an alternative to the processes of vector interpolation. With this approach, it is possible to find new elevation data from the information contained in DEM, and generate new data with the same spatial resolution.

The use of the three algorithms allows processing huge quantity of data contained in DEM for simulation and visualization processes, optimizing its performance.

## Acknowledgments

The authors of this paper wish to thank the CIC, CGPI, IPN and CONACyT for their support. Specially thanks the reviewers for their pertinent comments.

## References

1. Modelos Digitales de Elevación, Generalidades y Especificaciones, Instituto Nacional de Estadística, Geografía e Informática de México, Aguascalientes, México, 1999 (ISBN: 970-13-2511-7).
2. Normas Técnicas para la Elaboración de Ortofotos Digitales, Instituto Nacional de Estadística, Geografía e Informática de México, Aguascalientes, México, 1999 (ISBN: 970-13-2510-9).
3. M. Duchaineu, *Roaming Terrain: Real-time Optimally Adapting Meshes*, *Proceedings of the SIGGRAPH' 99*, Denver, USA, 1999, 56-67.
4. P. Lindstrom, *Real-time, Continuous Level of Detail Rendering of Height Fields*, *Proceedings of the SIGGRAPH 96*, Lyon, France, 1996, 89-101.
5. R. Quintero, *et al*, *Skeleton-based Algorithm for Increasing Spectral Resolution in Digital Elevation Model*, *Progress in Pattern Recognition, Image Analysis and Applications*, , Lecture Notes in Computer Science (LNCS), Vol. 3287, Springer-Verlag, Berlin Heidelberg, 2004, ISSN: 0302-9743, 550-557.
6. A. Watt, *Advanced Animation and Rendering Techniques*, (USA, Addison-Wesley, 1999).
7. A. Watt, *3D Computer Graphics*, (USA, Addison-Wesley, 2000).



# Object Image Retrieval by Shape Content in Complex Scenes Using Geometric Constraints

Agnés Borràs and Josep Lladós\*

Computer Vision Center - Dept. Informàtica, UAB Bellaterra 08193, Spain  
{agnesba, josep}@cvc.uab.es  
<http://www.cvc.uab.es>

**Abstract.** This paper presents an image retrieval system based on 2D shape information. Query shape objects and database images are represented by polygonal approximations of their contours. Afterwards they are encoded, using geometric features, in terms of predefined structures. Shapes are then located in database images by a voting procedure on the spatial domain. Then an alignment matching provides a probability value to rank de database image in the retrieval result. The method allows to detect a query object in database images even when they contain complex scenes. Also the shape matching tolerates partial occlusions and affine transformations as translation, rotation or scaling.

## 1 Introduction

The goal of Content-Based Image Retrieval (CBIR) is to find all images in a given database that contain certain visual features specified by the user. The reviews of Huang [1] and Forsyth [2] expose a wide variety of feature representations and image retrieval strategies. This work is focused on the development of a CBIR system where the image classification is done according to the shape information. Given the image of an object and a database containing images of complex scenes, the system is able to retrieve those images that likely contain an instance of the object.

In the literature we can find a great variety of shape representation approaches. Some relevant surveys are those of Veltkamp [4], Safar [5], Zhang [3] or Loncarnic[11]. Some retrieval strategies represent the shape taking the information of the whole image. This fact allows to obtain a compact representation that works efficiently in a retrieval application. This is the case of approaches that use shape descriptors such as the shape context [6], the Fourier coefficients or the ART descriptor of the standard MPEG7 [7]. Although these strategies provide relevant results they are not suitable for retrieving objects in complex scenes. In this case it is essential to apply a structural approach that permits to detect a shape as a part of the entire information of an image. Structural approaches on shape representation and matching often use graph based strategies [8][9][10].

---

\* This work has been partially supported by the project TIC2003-09291 and the grant 2002FI-00724.

In our work we have chosen a shape representation based on the boundary information. This fact allows us to deal with open shapes and sketches. Query shape objects and database images are represented by polygonal approximations of their contours. Afterwards they are encoded, using geometric features, in terms of predefined structures. Shapes are then located in database images by a voting procedure on the spatial domain. Then an alignment matching provides a probability value to rank de database image in the retrieval result. The method allows detecting a query object in database images even when they contain complex scenes. Also the shape matching tolerates partial occlusions and affine transformations of translation, rotation or scaling.

The stages of our method guide the structure of this paper. In the section 2 we present the shape modelling strategy and we detail the representation of an image. Then, in the section 3 we proceed to define how the modelling information is used in the detection process. Finally, in sections 4 and 5 we present the obtained results and the conclusions of our work.

## 2 Shape Modelling

A CBIR system analyses the similarity of a given image against a set of images contained in a database. The retrieval needs a previous step where the shape information is modelled. The modelling is applied in the same manner to the query image and to the database images, so we generalize the explanation for any image  $I$ .

### 2.1 Geometric Features of the Shape Elements

To model a shape we use the boundary information polygonally approximated in terms of segments. We assign each one a reference orientation, thus we refer them as vectors instead of segments.

**Definition 1.** *We define the boundary information  $BI$  of an image  $I$  as the basic data used to model a shape. Then we denote  $VI$  the collection of  $N$  vectors  $v$  that conform the polygonal approximation of  $BI$ .*

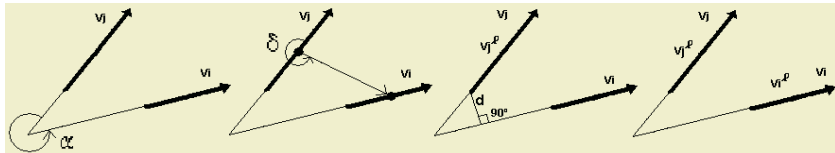
In Figure 3 we can see graphically the steps of the retrieval system. Then, from an image  $I$  we can see the  $BI$  and  $VI$  information. The vectors that compose a shape are identified in a unique way with a set of features called *absolute features*.

**Definition 2.**  *$AF(v)$  is defined as an attribute function that, given an image vector  $v$  assigns its length, angle and coordinates as absolute features. They are denoted  $AF^l(v)$ ,  $AF^\alpha(v)$  and  $AF^{(x,y)}(v)$  respectively.*

Notice that the absolute features contain the data that describe the scale, rotation and translation of a vector in an image. On the contrary, we want our system to be invariant to these three affine transformations. This way, instead

of working directly from the absolute features we consider them pairwise and extract what we call *relative features*. For the sake of simplicity, we will denote  $v_{ij}$  the vector pair  $(v_i, v_j)$ .

**Definition 3.** Given  $AF(v_i)$  and  $AF(v_j)$  we denote  $RF(v_{ij})$  the attribute function that computes the relative features for the vector pair  $v_{ij}$ . These features are the relative distance, the relative angle, the relative size, and the medium relative angle. We denote them  $RF^d(v_{ij})$ ,  $RF^\alpha(v_{ij})$ ,  $RF^l(v_{ij})$  and  $RF^\delta(v_{ij})$  respectively.



$$RF^\alpha(v_{ij}) = \alpha \quad RF^\delta(v_{ij}) = \delta \quad RF^d(v_{ij}) = \frac{d}{AF^l(v_i)} \quad RF^l(v_{ij}) = \frac{AF^l(v_j)}{AF^l(v_i)}$$

Fig. 1. Computation of the relative features

The relative features of two vectors can be seen as basic shape descriptors (the Figure 1 shows its computation graphically). In a higher abstraction level, a shape is described in term of primitives that combine these low level descriptors.

### 2.2 Labelling of the Image Elements with the Primitive Structures

Many shape recognition strategies search for particular line arrangements according to perceptual grouping of salient features [13][12][15]. In our case, we describe the relationships of perpendicularity, parallelism and co linearity due to several predefined structures that we call primitives.

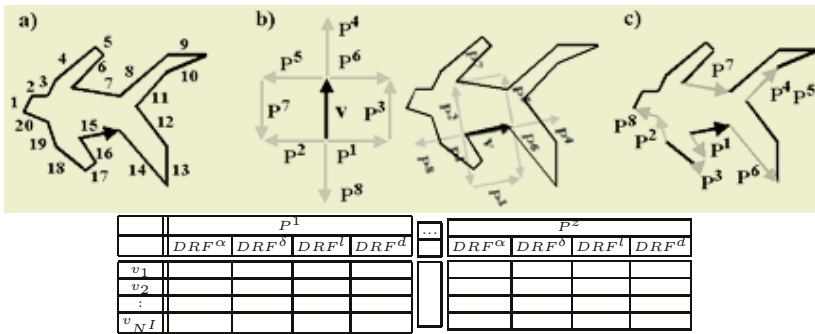
**Definition 4.** A primitive  $P$  is a particular arrangement of two vectors  $(w_a, w_r)$ . We denote  $\mathcal{P}$  a collection of primitives  $\{P^z\} z = 1..N^{\mathcal{P}}$

Figure 3 shows the set of primitive types we consider due to the arrangement of the vectors  $w_a$  and the reference vector  $w_r$  (shown in black). The aim of the shape modelling consists in obtaining a local description of the shape around each image vector. The idea is easily shown in the Figure 2. We identify the reference vectors  $w_r$  of every primitive on an image vector  $v_i$ . Then for every vector  $w_a$  we have to find the most similar image vector  $v_j$ . We choose  $v_j$  to be the most similar if it minimizes a given distance function  $DRF$ .

**Definition 5.** Given the relative features of an image vector pair,  $RF(v_{ij})$ , and the relative features of a primitive vector pair,  $RF(w_{ra})$ , let us denote  $DRF^\alpha$ ,  $DRF^\delta$ ,  $DRF^d$ ,  $DRF^l$  the distance values for each relative feature and  $DRF$  the function that globally quantifies the distance from  $v_{ij}$  to  $w_{ra}$ .

$$DRF(RF(v_{ij}), RF(w_{ra})) = \max(DRF^\alpha, DRF^\delta, DRF^d, DRF^l)$$

The election of the greatest distance for each feature assures that DRF provides a certain degree of similarity only when all the features accomplish at least this similarity degree. The general idea to compute every DRF value consists in find the difference between the relative feature values and normalize the result by the maximum feasible variation. Every shape element  $v_i$  is assigned an attribute vector composed by the distances of the relative features in relation to every primitive. This information is arranged in a table  $LI$  where the rows are referred to the vectors and the columns are referred to the primitives (see Figure 2). In other words, every line of  $LI$  can be understood as the deformation that the primitive structures have to suffer to fit locally the shape around  $v_i$ .



**Fig. 2.** a) Image vectors b) Arrangement of the primitive vectors  $w_a$  due to the identification of  $v_{15}$  with every  $w_r$  c) The labelling process searches the most similar vectors to every  $w_a$  (ex:  $w_a$  of belonging  $P^1$  is identified with  $v_{16}$ ). The modelled information is indexed  $L[i][z]$  in the table

### 3 Shape Detection

Given a query image  $I_1$  we evaluate the retrieval of a database image  $I_2$  in a two step process.

#### 3.1 Location of the Shape: The Voting Process

The shape detection involves a voting procedure in the spatial domain that uses the modelled information of both images and a reference point  $rp$  in the image  $I_1$ . The evidence combination methods that share these characteristics are typically those based in the generalised Hough transform [16][17].

**Definition 6.** Let us define a vote  $m_{ijO}$  as the evaluation of the local matching of the vector  $v_i$  belonging to  $I_1$  with the vector  $v_j$  belonging to  $I_2$  in the orientation  $O$  (where  $O=1$  means equal orientation and  $O=0$  means opposite orientation).

Given two modelled images, the process generates  $N_1 \times N_2 \times 2$  votes. The votes are used to construct an image map  $M$  of the shape location.

**Definition 7.** Let us name  $M$  an image with the same dimensions as  $I_2$  that acts as probability map of the locations of the shape  $I_1$  inside  $I_2$ .

Every vote has a specific location  $L(m_{ijO})$  in the map  $M$ . This location is found by the transformation of the reference point  $rp$  when we match  $v_i$  and  $v_j$ . Moreover, every vote has a weight  $H(m_{ijO})$  that is computed from the information  $LI_1[i]$  and  $LI_2[j]$ . This information describes the distortions of the shapes around the vectors  $v_i$  and  $v_j$  respect to the primitives. The query shape is detected when the local distortions are similar to those of the database shape. This way, the vote weight  $H(m_{ijO})$  will have a high value if  $LI_1[i]$  and  $LI_2[j]$  are similar. When the vote evaluates  $v_i$  on  $v_j$  with the same orientation ( $O=0$ ) we analyse pairwise the information of  $L[i]$  and  $L[j]$  for each primitive. Otherwise, we compare the information related to the primitives that have the same characteristics but opposite orientation ( $P^1$  with  $P^5$ ,  $P^2$  with  $P^6$ , and so on). A vote with a high weight enforces the probability of finding the query shape in the location  $L(m_{ijO})$ . The map  $M$ , viewed in a 3D representation, shows as peaks the locations where the query shape is more probably located (see the example Figure 3). Then we proceed to validate the shape detection on those positions such  $M(x, y)$  exceeds a certain confidence value  $Thr_M$ .

### 3.2 Retrieval Evaluation: The Alignment Process

The generated votes are accumulated with freedom of scale a rotation as peaks in  $M$ . Then we validate its coherence using an alignment process on the original contour information of both images,  $BI_1$  and  $BI_2$ . Given a vote  $m_{ijO}$ , the alignment points are defined by the initial and final coordinates of  $v_i$  and  $v_j$ . We evaluate the matching by combining the spatial distance between the contours with and the angular information of the normal vectors on shape boundary [17][18]. Then, we use the maximum alignment result of all the peaks in  $M$  as a measure to rank the database images in the retrieval process (see Figure 3).

## 4 Results

We have test our CBIR system with three experiments. The first one is composed by 75 database images and 5 query images of logos. The second deals with 48 database images and 3 possible queries of traffic signs [19]. Finally we have used another set of 30 images and 6 image queries of tin cans. For every query image we have computed the rate of database images that contain the searched object and that have been retrieved in the first  $n$  positions (being  $n$  the total amount of database images where the query shape can be found). The obtained results for the three tests are 97%, 80%, and 65%. The images of the Figure 4 show the performance of the shape retrieval against transformations of rotation and scale. The traffic sign test shows the robustness of the algorithm against a great amount of information related to the scene. Furthermore, the first image of the third test shows the tolerance of the system against partial occlusion. We have to stand

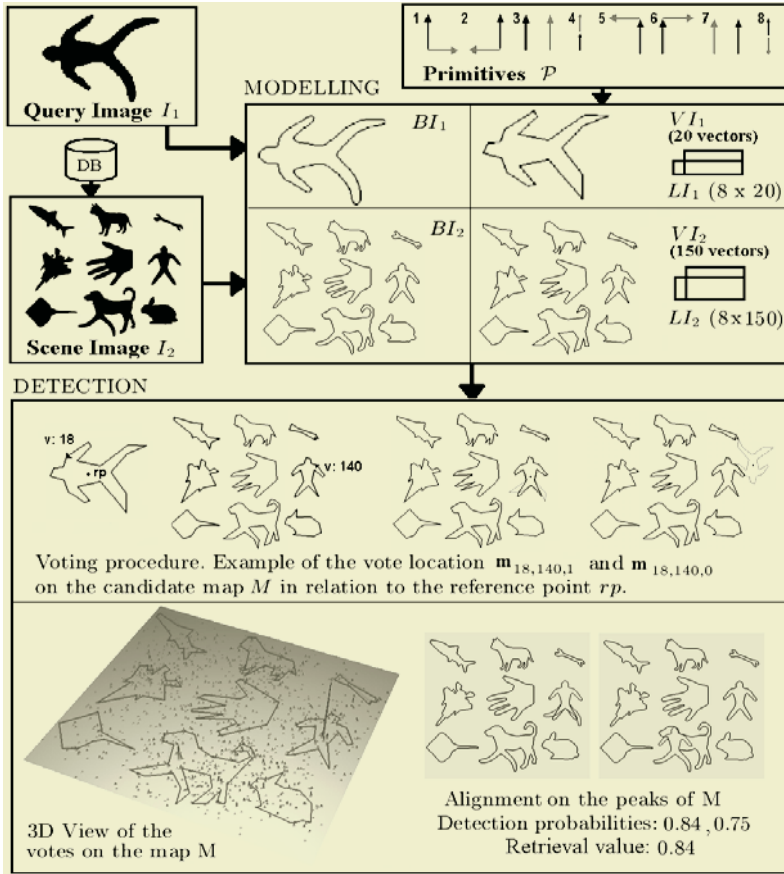
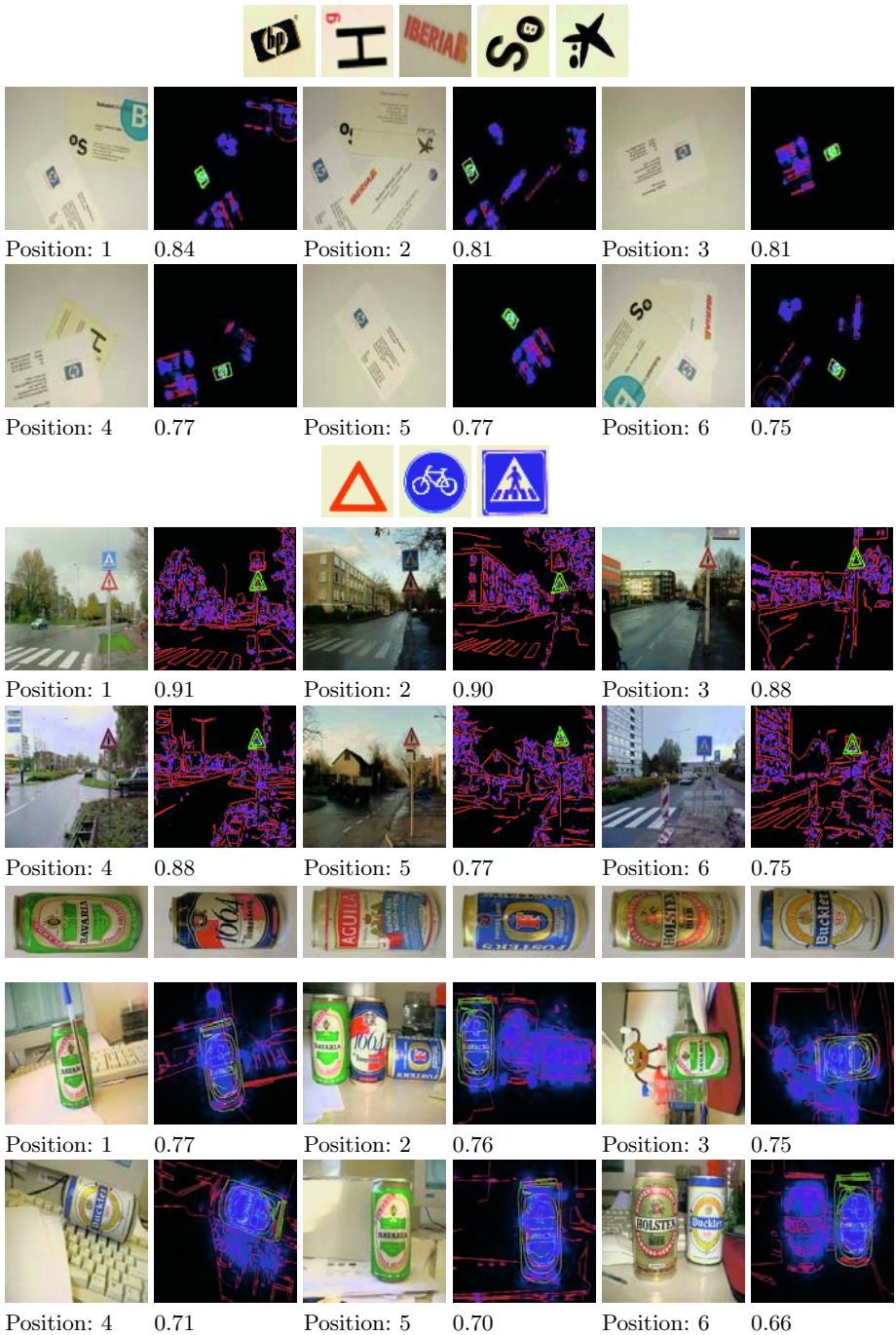


Fig. 3. Example of the algorithm steps

out that the object retrieval in real scenes, such as the tin can example, adds the difficulty of dealing with the effects such as specular reflections, shades or slightly modifications of the viewpoint. These effects are captured by the contour information and affect directly to the vector structures and to the alignment result. Finally we have experimentally set at 0.75 the confidence threshold on the retrieval results as a compromise between the precision and the recall.

## 5 Conclusions

We have developed a CBIR system of 2D objects by shape content that is capable to deal with databases of complex scenes. The system is modularised in two blocs: the shape modelling and the shape detection. The independence of the two parts allows to precompute the shape representation for any database image. The algorithm has been tested in real scenes to evaluate the robustness of the object location against transformations of scale, rotation and partial occlusions.



**Fig. 4.** Retrieval examples: Traffic signs, logos and beer cans. The first line corresponds to the query images. We can see the first 6 results on the leftmost query image, the vectorized image that shows the location solution, and the retrieval value

## References

1. Huang, T., Rui, Y.: Image retrieval: Past, present, and future. International Symposium on Multimedia Information Processing, 1997
2. Forsyth, D.A., Malik, J., Fleck, M.M., Greenspan, H., Leung, T.K., Belongie, S., Carson, C., Bregler, C.: Finding Pictures of Objects in Large Collections of Images, Object Representation in Computer Vision (1996) 335-360
3. Zhang, D., Lu, G.: Review of shape representation and description techniques, PR Volume 37, No 1, 1-19, Jan. 2004
4. Veltkamp, R., Hagedoorn, M.: State-of-the-art in shape matching, UU-CS-1999-27, Utrecht University, the Netherlands, 1999
5. Safar, M., Shahabi C., Sun, X.: Image Retrieval by Shape: A Comparative Study, IEEE International Conference on Multimedia and Expo (I), 141-144, 2000
6. Belongie, S., Malik, J.: Matching with shape context, IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL) 2000
7. B.S. Manjunath, P. Salembier, T. Sikora, editors: Introduction to MPEG-7: Multimedia Content Description Interface. Wiley, 2002
8. Huet, B., Cross, A., Hancock, E.R.: Shape Retrieval by Inexact Graph Matching, ICMCS, Volume 1, 772-776, 1999
9. Lourens T., Rolf P. Würtz: Object Recognition by matching symbolic edge graphs, Proc. on the Third Asian Conf.on CV, Hong Kong, 8 - 11, 1998
10. Bunke, H.: Inexact Graph Matching for Structural Pattern Recognition, PRL, Volume 1, No 4, "245-253", 1983
11. Loncaric, S.: A survey of shape analysis techniques, PR, Volume 31, No 8, 983-1001, 1998
12. Stein, F., Medioni, G: Structural indexing: Efficient 2D object recognition, IEEE Transactions on PAMI, Volume 14 , Issue 12 , 1198 - 1204, Dec. 1992
13. Eakins, JP., Shields, K., Boardman, J.: ARTISAN – a shape retrieval system based on boundary family indexing, Storage and Retrieval for Still Image and Video Databases IV. Proceedings SPIE 2670, 17-28, 1996
14. Hu, J., Pavlidis, T.: A hierarchical approach to efficient curvilinear object searching, Computer Vision and Image Understanding Volume 63, Issue 2 , 208-220, 1996
15. Etemadi, A. , Schmidt, J.P., Matas, G., Illingworth, J., Kittler, J.V.: Low-Level Grouping of Straight Line Segments, BMVC91, 119-126, 1991
16. Lee, H.M., Kittler, J.V., Wong, K.C.: Generalised Hough Transform in Object Recognition, ICPR92, 285-289, 1992
17. Gonzalez-Linares, J.M, Guil, N., Zapata, E.L.: An Efficient 2D Deformable Objects Detection and Location Algorithm, PR, Volume 36, Issue 11, 2543-2556, Nov. 2003
18. Huttenlocher, D.P., Ullman, S.: Object Recognition Using Alignment IUW, Volume 87, 370-380
19. Grigorescu, C., Ptekov, N.: Distance Sets for Shape Filters and Shape Recognition, IEEE Transactions on Image Processing, Volume 12, No 10, 1274-1286, Oct. 2003



## Part III

# Image and Video Processing

# A Real-Time Gabor Primal Sketch for Visual Attention\*

Alexandre Bernardino and José Santos-Victor

IST, Instituto de Sistemas e Robótica – Lisboa  
{alex,jasv}@isr.ist.utl.pt  
<http://vislab.isr.ist.utl.pt>

**Abstract.** We describe a fast algorithm for Gabor filtering, specially designed for multi-scale image representations. Our proposal is based on three facts: first, Gabor functions can be decomposed in gaussian convolutions and complex multiplications which allows the replacement of Gabor filters by more efficient gaussian filters; second, isotropic gaussian filtering is implemented by separable 1D horizontal/vertical convolutions and permits a fast implementation of the non-separable zero-mean Gabor kernel; third, short FIR filters and the *à trous* algorithm are utilized to build a recursive multi-scale decomposition, which saves important computational resources. Our proposal reduces to about one half the number of operations with respect to state-of-the-art approaches.

## 1 Introduction

Gabor filtering is widely applied in image analysis and computer vision applications, such as image compression [5], texture classification [14], image segmentation [15], motion analysis [1] and visual attention [8]. The use of Gabor filters is motivated by information theoretic and biological facts. Gabor [6] showed that gaussian-modulated complex exponentials provide the best trade-off between spatial and frequency resolution. Neurophysiological studies show that visual cortex simple cells are well modeled by families of 2D Gabor functions [4]. Both facts raised considerable interest and suggest that neuronal structures may develop toward optimal information coding.

In the case of visual attention, recent models propose multi-scale image representations of different features like color, intensity and orientation [8]. Such a decomposition benefits, in terms of completeness and stability, on having more than one voice (frequency) per scale and orientation [11]. Therefore, a large number of different kernels may be needed to represent the image characteristics.

Whereas fast algorithms for Gabor filtering exist [13, 18], multi-scale representations require analysis with many Gabor kernels, tuned to different orientations, scales and frequencies, which poses serious computational constraints in real-time scenarios. However, many computations are redundant. Here we exploit this redundancy to develop more efficient algorithms.

---

\* Research partially funded by European Project IST 2001 37540 (CAVIAR).

In section 2 we review some of the underlying theory of Gabor analysis and show that image filtering with isotropic zero-mean Gabor kernels (non-separable) can be computed by the sum of two separable filtering operations. In section 3 we show that Gabor filtering can be factored in complex multiplications and gaussian convolutions, which allow significant computational improvements. In section 4 we apply this technique to multi-scale image analysis and propose an approximate algorithm that reduces computations more than 50%.

## 2 Isotropic Gabor Wavelets

Gabor functions consist on the multiplication of a complex exponential (carrier) and a gaussian function (envelope). We will focus on isotropic envelope functions because efficient separable implementations are currently available. Let  $w_\sigma(x, y)$  be a two dimensional gaussian function with scale  $\sigma$  and,  $c_\psi(x, y)$ ,  $\psi = (\lambda, \theta)$  be a complex exponential function representing a plane-wave with wavelength  $\lambda$  and orientation  $\theta$ :

$$w_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{and} \quad c_\psi(x, y) = e^{i\frac{2\pi}{\lambda}(x \cos \theta + y \sin \theta)}$$

To simplify notation, we will drop the spatial coordinates  $(x, y)$  and write a two dimensional Gabor function as  $\mathbf{g}_{\sigma,\psi} = \mathbf{w}_\sigma \cdot \mathbf{c}_\psi$ . This function has non zero mean value, which is not desirable for the purpose of feature extraction and multi-scale analysis. The zero-mean kernel is used instead [11]:

$$\gamma_{\sigma,\psi} = \mathbf{w}_\sigma \cdot (\mathbf{c}_\psi - k_{\sigma,\psi}) \tag{1}$$

where the scalar  $k_{\sigma,\psi}$  is calculated so the kernel’s average value is zero (Appendix A). We distinguish between the **Gabor function** (non-zero-mean function) and the **Gabor kernel** (zero-mean function). The Gabor kernel satisfies the admissibility condition for wavelets, thus being suited for multi-resolution analysis [12]. Apart from a scale factor, it is also known as the Morlet Wavelet. Examples of two dimensional Gabor kernels are shown in Figure 2.

Image analysis by convolution with Gabor kernels has been extensively studied in the literature. In practical terms, the filter will respond strongly when the local image structure is similar to the Gabor kernel shape, in terms of scale ( $\sigma$ ), wavelength ( $\lambda$ ), and orientation ( $\theta$ ). Using the definition of the Gabor kernel (1), its convolution with an image  $\mathbf{f}$  can be written as:

$$\mathbf{z}_{\sigma,\psi} = \underbrace{\mathbf{f} * \mathbf{g}_{\sigma,\psi}}_{\mathbf{z}_{\sigma,\psi}^c} - k_{\sigma,\psi} \underbrace{\mathbf{f} * \mathbf{w}_\sigma}_{\mathbf{z}_\sigma^w} \tag{2}$$

This convolution can be implemented by subtracting two terms:  $\mathbf{z}_{\sigma,\psi}^c$  - a Gabor convolution; and  $k_{\sigma,\psi} \mathbf{z}_\sigma^w$  - a scaled gaussian convolution. In the isotropic case both Gabor and gaussian functions are separable ( $g(x, y) = g_x(x) \cdot g_y(y)$ , and  $w(x, y) = w_x(x) \cdot w_y(y)$ ) and convolutions can be performed with two cascaded (horizontal and vertical) 1D convolutions. Thus, even though the isotropic Gabor

kernel  $\gamma$  is not separable itself (can not be written as the tensor product of two 1D filters), image filtering with this kernel can be implemented efficiently as the sum of two separable convolutions.

To date, the fastest implementation of gaussian [17] and Gabor convolutions [18] require 13 real (gaussian) and 13 complex (gabor) arithmetic operations per pixel per dimension. Considering a complex multiplication as 4 real multiplicationa and 2 real additions, the extension to 2-D signals requires, respectively, 26 and 108 real operations. Therefore, image convolution with Gabor kernels,consisting in 1 gaussian filtering, 1 Gabor filtering, 1 multiplication and 1 addition, has a total computational cost of 136 operations per pixel.

### 3 Gabor Convolution Factorization

We show that the Gabor convolution in (2) can be computed by multiplications with complex exponentials and gaussian convolutions. The motivation is that state-of-the-art gaussian filtering is significantly more efficient than Gabor filtering. We focus on the isotropic case but the method can also be applied to the anisotropic case. In fact, a separable implementation of anisotropic Gabor filtering has recently been proposed [7].

Image convolution with Gabor functions, denoted by  $\mathbf{z}_{\sigma,\psi}^c$ , is computed by:

$$z_{\sigma,\psi}^c(x, y) = \sum_{k,l} f(k, l) \cdot w_{\sigma}(x - k, y - l) \cdot c_{\psi}(x - k, y - l)$$

Since  $\mathbf{c}_{\psi}(x - k, y - l) = c_{\psi}(x, y)\bar{c}_{\psi}(k, l)$  ( $\bar{c}$  denotes complex conjugation), we can expand the previous expression into:

$$z_{\sigma,\psi}^c(x, y) = c_{\psi}(x, y) \cdot \sum_{k,l} \bar{c}_{\psi}(k, l) \cdot f(k, l) \cdot w_{\sigma}(x - k, y - l)$$

In compact form, the full convolution (2) is written as:

$$\mathbf{z}_{\sigma,\psi} = \underbrace{\mathbf{c}_{\psi} \cdot [(\mathbf{f} \cdot \bar{\mathbf{c}}_{\psi}) * \mathbf{w}_{\sigma}]}_{\mathbf{z}_{\sigma,\psi}^c} - k_{\sigma,\psi} \cdot \underbrace{(\mathbf{f} * \mathbf{w}_{\sigma})}_{\mathbf{z}_{\sigma}^w} \tag{3}$$

Notice that only gaussian convolutions are involved in the previous expression. With the IIR gaussian filter of [17] (26 real operations per pixel), the required computations on Eq. (3), are:

- a **modulation** ( $\mathbf{f} \cdot \bar{\mathbf{c}}_{\psi}$ ) corresponding to **2 operations** per pixel;
- a **complex gaussian filtering** ( $\mathbf{w}_{\sigma}$  convolved with  $\mathbf{f} \cdot \bar{\mathbf{c}}_{\psi}$ ) requires **52 operations** per pixel;
- a **demodulation** operation (product of  $\mathbf{c}_{\psi}$  with  $(\mathbf{f} \cdot \bar{\mathbf{c}}_{\psi}) * \mathbf{w}_{\sigma}$ ) requires 1 complex multiplication per pixel, corresponding to **6 operations** per pixel;
- a **real gaussian filtering** ( $\mathbf{f} * \mathbf{w}_{\sigma}$ ) requiring **26 operations** per pixel;
- a **real scaling** by  $k_{\sigma,\psi}$ , requires **1 operation** per pixel;
- and the **final subtraction**, corresponds to only **1 operation** per pixel because only the real part of Gabor kernels has non zero DC value.

Altogether we have 88 operations which, in comparison with the reference value of 136 operations, correspond to about **35%** savings in computation.

When multiple carriers (orientations/wavelengths) are considered, it is obvious from Eq. (3) that term  $\mathbf{z}_\sigma^w$  is common to all. Fig. 1 shows a graphical representation of the method. Gaussian filtering contributes with 26 operations and each carrier contributes with additional 62 operations (our proposal) or 110 operations (direct Gabor filtering). If, for example, 4 orientations and 2 wavelengths are used, the total number of operations is  $8 \times 62 + 26 = 522$  vs  $8 \times 110 + 26 = 906$  (about **42%** savings). It is also worth mentioning that multi-scale image architectures often compute image gaussian expansions to support further processing[2, 3]. Thus intermediate filtered images  $\mathbf{z}_\sigma^w$  may already have been computed, which saves additional 26 operations per pixel.

## 4 Analysis at Dyadic Scales

Dyadic scale representations are very utilized in image analysis. Efficient recursive algorithms exist to build Gaussian and Laplacian pyramids [2] with  $L$  dyadic levels ( $\sigma \approx 2^i, i = 0, \dots, L$ ). Usual approaches create image pyramids by successively filtering previous levels and sub-sampling by 2. Even though sub-sampling is useful in terms of storage and computation, it has the disadvantage of losing translation invariance properties [12], thus reducing precision in the localization of relevant image structures. We consider the unsubsampled case, where image size is constant at all scales. In this case the *à trous* algorithm [12], is an efficient recursive technique to implement multi-resolution decompositions with constant size filters. If filter coefficients are properly chosen, we obtain good approximations to quasi-dyadic gaussian filters [2].

Consider a signal  $f(x, y)$  and low-pass filter  $q(x, y)$  with Fourier transform  $\tilde{q}(\omega_x, \omega_y)$ . The first step of the *à trous* algorithm consists in obtaining a low-pass version of the original signal:  $\mathbf{f}^1 = \mathbf{f} * \mathbf{q}$ . In the next decomposition level a new filter is created by expanding the previous one with zero insertion, which, in the frequency domain, corresponds to a spectral compression  $\tilde{q}^1(\omega_x, \omega_y) = \tilde{q}(2\omega_x, 2\omega_y)$ . The new low-pass signal is computed by  $\mathbf{f}^2 = \mathbf{f}^1 * \mathbf{q}^1$ , and the procedure goes on recursively until the last scale level is reached. Since the convolution operation is linear, this is equivalent to filter the original signal  $\mathbf{f}$  with filters  $\mathbf{w}^i$  resulting from successive convolutions of the several  $\mathbf{q}^k$ . In the Fourier domain the equivalent filters are described by  $\tilde{w}^i(\omega_x, \omega_y) = \prod_{k=0}^i \tilde{q}(2^k \omega_x, 2^k \omega_y)$ . In [2], some base filters  $\mathbf{q}$  were tested but not all choices approximate gaussian functions. We use the 1D filter  $q_x = (.05, .25, .40, .25, .05)$  for  $x = (-2, -1, 0, 1, 2)$  to generate a set of equivalent filters similar to dyadic gaussian functions.

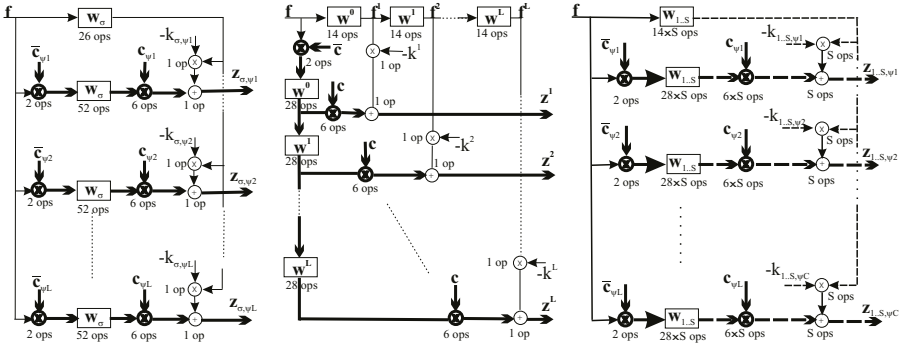
Since the filter is symmetric, convolution is computed in the following way:

$$f^{i+1}(\cdot) = q_0 f^i(\cdot) + q_1 [f^i(\cdot - 2^i) + f^i(\cdot + 2^i)] + q_2 [f^i(\cdot - 2^{i+1}) + f^i(\cdot + 2^{i+1})]$$

In this form, only 6 multiplications and 8 additions per pixel are required to perform the 2D convolution. For a single carrier, we can compute a multi-scale approximation to (3) with 52 operations in the first level and 50 operations in

the remaining levels (see Fig. 1). This corresponds to **62%** computation savings with respect to the reference value of 136 operations per pixel.

Finally, let us consider the multi-scale, multi-carrier problem. If  $S$  is the number of scales and  $C$  the number of carriers, our proposal requires  $14 \times S + 2 \times C + 36 \times S \times C$  operations (see Fig. 1), while direct Gabor filtering requires  $26 \times S + 2 \times C + 108 \times S \times C$  operations. If all combinations of carriers and scales are needed, then we attain up to **66%** computation savings. For example, considering 3 scales ( $S = 3$ ), 4 orientations and 2 wavelengths ( $C = 8$ ), the full decomposition takes 922 operations *vs* 2686 operations in the reference method.



**Fig. 1.** Proposed Gabor filtering schemes: single-scale-multi-carrier (left), multi-scale-single-carrier (middle) and multi-scale-multi-carrier (right). Thick/Thin lines and boxes represent complex/real signals and filters. At each computational element we indicate the number of real operations required. Dashed lines represent vectors instead of scalars.

## 5 A “Real-Time” Multi-scale Quasi-Gabor Expansion

We have developed a quasi-dyadic Gabor image decomposition for the control of visual attention in an active vision system, using the the first 4 scales generated by the *à trous* algorithm  $\sigma = \{0.95, 2.12, 4.35, 8.75\}$ . The definition of the carrier wavelengths,  $\lambda$ , is inspired on biological data. Simple and complex cells in the primary visual cortex have receptive fields that resemble Gabor functions of particular combinations and ranges of parameters [11]. In particular the half-amplitude frequency bandwidth ( $\beta$ ) range from 0.5 to 2.5 octaves. In the radial frequency direction, an isotropic Gabor function is given by  $\tilde{\mathbf{g}}(\omega) = e^{-\frac{1}{2}\sigma^2(\omega - \frac{2\pi}{\lambda})^2}$ , whose half-amplitude points  $\omega_{1,2}$  and half-amplitude bandwidth  $\beta$  are, in octaves:

$$\omega_{1,2} = \frac{2\pi}{\lambda} \pm \frac{\sqrt{2\log(2)}}{\sigma} \quad \text{and} \quad \beta = \log_2 \frac{2\pi\sigma + \lambda\sqrt{2\log(2)}}{2\pi\sigma - \lambda\sqrt{2\log(2)}}$$

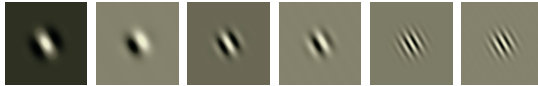
We have used wavelength values  $\lambda = \{3.7, 7.4, 14.8, 29.6\}$ . The half-amplitude bandwidth of each scale/wavelength combination is shown in table 1. We choose kernels whose half-amplitude bandwidth is approximately within biologically

**Table 1.** Half-amplitude bandwidth (in octaves) for each pair scale/wavelength. Italic entries are biologically plausible values. In parenthesis we indicate the appearance of the kernel: E – “edge” kernel, ST – small texture kernel, LT – large texture kernel.

	$\lambda = 3.7$	7.4	14.8	29.6
$\sigma = .95$	<i>2.68 (E)</i>	-	-	-
2.12	<i>.98 (ST)</i>	<i>2.26 (E)</i>	-	-
4.35	<i>.46 (LT)</i>	<i>.95 (ST)</i>	<i>2.18 (E)</i>	-
8.75	<i>.23</i>	<i>.46 (LT)</i>	<i>.95 (ST)</i>	<i>2.16 (E)</i>

**Table 2.** Signal to Error Ratio (in dB) between the output of FIR Gabor wavelets and the proposed approximation. Test images are from the collections *miscellaneous*, *aerial* and *texture* of the USC-SIPI database.

SNR	Aerial	Texture	Misc
Average	30.39	30.06	29.95
Maximum	38.87	39.28	38.92
Minimum	23.82	13.83	7.15



**Fig. 2.** Real and imaginary parts of: (left) an “edge” (E) Gabor kernel with half-frequency bandwidth in octaves  $\beta = 2.46$ ; (center) a “small texture” (ST) kernel having  $\beta = 1.04$ ; (right) a “large texture” (LT) kernel with  $\beta = 0.51$ .

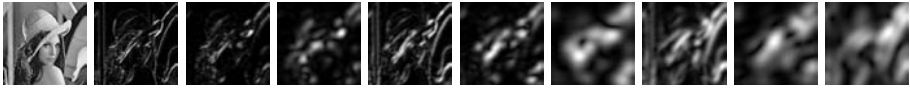
plausible values (italic entries in table 1). The kernel shapes are shown in Fig. 2, and resemble units tuned to edges, small texture patches and large texture patches, respectively. Roughly speaking, “edge” kernels will respond equally well in image locations corresponding to edges and textures with appropriate scale and orientation. “Texture” kernels will respond better in textured areas with the matched direction and wavelength.

Notice that not all combinations of wavelengths and scales are biologically plausible. A recursive dyadic decomposition will require  $14 \times S + 2 \times C + 28 \times A_k + 6 \times R_k$  operations, where  $A_k$  is the number of levels to compute and  $R_k$  is the number of “interesting” kernels. With the IIR filters, some levels are not required and the number of operations is  $26 \times S + 2 \times C + 52 \times R_k + 6 \times R_k$ . In the proposed decomposition, we have  $A_k = 60$  and  $R_k = 36$ , which lead to 1984 operations in the dyadic recursive decomposition and 2224 with IIR gaussian filters. For the sake of comparison, if the state-of-the art IIR Gabor filters are used, the number of computations would increase to 4024 operations.

## 6 Results

Figure 3 shows the output modulus of the proposed filter, applied to a common test image. The computation, in  $128 \times 128$  greyscale images, takes about 0.2 seconds in a P4 2.66GHz processor.

We have applied both the approximate method (with the *à trous* decomposition) and the exact method (with FIR Gabor wavelets) to images from the *miscellaneous*, *aerial* and *texture* classes [16], converted to greyscale and  $128 \times 128$  pixel sizes. We have applied a decomposition of the type described in section 5,



**Fig. 3.** Modulus of the Gabor wavelet decomposition (for orientation  $135^\circ$ ) applied to the image “Lenna” (left). Contrast has been normalized for visualization. From left to right, the kernel parameters  $(\sigma, \lambda)$  are, respectively:  $(0.95, 3.7)$ ,  $(2.12, 3.7)$ ,  $(4.35, 3.7)$ ,  $(2.12, 7.4)$ ,  $(4.35, 7.4)$ ,  $(8.75, 7.4)$ ,  $(4.35, 14.8)$ ,  $(8.75, 14.8)$ ,  $(8.75, 29.6)$ .

with 4 orientations, and the relative mean squared error between the two methods was computed for all images and filter channels. On average, the signal to error ratio is about 30dB (3% error). In some images with strong edges in the boundary, the error grows larger (7dB), but current work is dealing with efficient boundary conditions to address this problem.

## 7 Conclusions

We have presented a novel algorithm for the computation of Gabor features. Improvements are obtained by an efficient decomposition of Gabor convolution into gaussian convolutions and complex multiplications, and the reuse of intermediate computations in a multi-scale framework. The method reduces computations to about one half when compared to the state-of-the-art. The application of Gabor filters is far from being limited to visual attention. One can find Gabor analysis in object representation [9] texture classification [10], motion estimation [1] and image compression [5]. Therefore, many other applications may benefit from the results presented in this paper.

## Appendix A Computation of Gabor Kernel’s $k$ Parameter

A Gabor Kernel is defined in the frequency domain as:

$$\tilde{g}(\omega_x, \omega_y) = \tilde{w} \left( \omega_x - \frac{2 \cos \theta}{\lambda \pi^{-1}}, \omega_y - \frac{2 \sin \theta}{\lambda \pi^{-1}} \right) - k \tilde{w}(\omega_x, \omega_y)$$

Parameter  $k$  is computed such that the kernels’ DC value is zero.

$$k = \frac{\tilde{w} \left( -\frac{2 \cos \theta}{\lambda \pi^{-1}}, -\frac{2 \sin \theta}{\lambda \pi^{-1}} \right)}{\tilde{w}(0, 0)}$$

With the *à trous* algorithm, the equivalent envelope filters  $\tilde{q}^i(\omega_x, \omega_y)$  have the following Fourier transform :

$$\prod_{k=0}^i (a \cos(2^{k+1} \omega_x) + b \cos(2^k \omega_x) + c) \cdot (a \cos(2^{k+1} \omega_y) + b \cos(2^k \omega_y) + c)$$



where  $a = 0.1$ ,  $b = 0.5$  and  $c = 0.4$ . Thus, the value of  $k$  comes:

$$\prod_{k=0}^i \left( a \cos \frac{2^k \cos \theta}{4\pi^{-1}\lambda} + b \cos \frac{2^k \cos \theta}{2\pi^{-1}\lambda} + c \right) \cdot \left( a \cos \frac{2^k \sin \theta}{4\pi^{-1}\lambda} + b \cos \frac{2^k \sin \theta}{2\pi^{-1}\lambda} + c \right)$$

## References

1. E. Bruno and D. Pellerin. Robust motion estimation using gabor spatial filters. In *Proc. of the 10th European Signal Processing Conference*, September 2000.
2. P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, 4(31):532–540, April 1983.
3. J. Crowley, O. Riff, and J. Piater. Fast computations of characteristic scale using a half-octave pyramid. In *CogVis 2002, International Workshop on Cognitive Computing*, Zurich, October 2002.
4. J.G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20:847–856, 1980.
5. S. Fischer and G. Cristóbal. Minimum entropy transform using gabor wavelets for image compression. In *Proc. of Int. Conf. on Image Analysis and Processing*, Palermo, Italy, September 2001.
6. D. Gabor. Theory of communication. *J. IEE*, 93:429–459, 1946.
7. Jan-Mark Geusebroek, Arnold W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. In *ECCV (1)*, pages 99–112, 2002.
8. L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of attention. *Vision Research*, 40:1489–1506, 2000.
9. V. Krueger and G. Sommer. Gabor wavelet networks for object representation. In *DAGM Symposium*, Kiel, Germany, September 2000.
10. P. Kruizinga, N. Petkov, and S.E. Grigorescu. Comparison of texture features based on gabor filters. In *Proc. of the 10th Int. Conf. on Image Analysis and Processing*, pages 142–147, Venice, Italy, September 1999.
11. T. Lee. Image representation using 2d gabor wavelets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(10), October 1996.
12. S. Mallat. *A Wavelet Tour of Signal Processing, 2nd Ed.* Academic Press, 1999.
13. O. Nestares, R. Navarro, and J. Portilla. Efficient spatial-domain implementation of a multiscale image representation based on gabor functions. *Journal of Electronic Imaging*, 7(1):166–173, January 1998.
14. T. Randen and Husøy. Image representation using 2d gabor wavelets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):291–310, April 1999.
15. T. Tangsukson and J.P. Havlicek. Am-fm image segmentation. In *Proc. IEEE Int. Conf. on Image Processing*, pages 104–107, Vancouver, Canada, September 2000.
16. Signal University of Southern California and Image Processing Institute. The usc-sipi image database. <http://sipi.usc.edu/services/database>.
17. I. Young and L. van Vliet. Recursive implementation of the gaussian filter. *Signal Processing*, 44:139–151, 1995.
18. I. Young, L. van Vliet, and M. van Ginkel. Recursive gabor filtering. *IEEE Trans. on Signal Processing*, 50(11):2798–2805, 2002.

# Bayesian Reconstruction of Color Images Acquired with a Single CCD

Miguel Vega<sup>1,\*</sup>, Rafael Molina<sup>2</sup>, and Aggelos K. Katsaggelos<sup>3</sup>

<sup>1</sup> Dpto. de Lenguajes y Sistemas Informáticos  
Universidad de Granada, Granada 18071, Spain  
mvega@ugr.es

<sup>2</sup> Dpto. de Ciencias de la Computación e Inteligencia Artificial  
Universidad de Granada, Granada 18071, Spain  
rms@decsai.ugr.es

<sup>3</sup> Department of Electrical and Computer Engineering  
Northwestern University, Evanston, IL 60208, USA  
aggk@ece.nwu.edu

**Abstract.** Most of the available digital color cameras use a single Coupled Charge Device (*CCD*) with a Color Filter Array (*CFA*) in acquiring an image. In order to produce a visible color image a demosaicing process must be applied, which produces undesirable artifacts. This paper addresses the demosaicing problem from a superresolution point of view. Utilizing the Bayesian paradigm, an estimate of the reconstructed images and the model parameters is generated.

## 1 Introduction

Most of the available digital color cameras use a single Coupled Charge Device (*CCD*) with a Color Filter Array (*CFA*) to obtain color images. Unfortunately the color filter generates different spectral responses at every CCD cell. The most widely used CFA is the Bayer one [1]. It imposes a spatial pattern of two G cells, one R, and one B cell, as shown in Fig. 1.

Bayer camera pixels convey incomplete color information which needs to be extended to produce a visible color image. Such color processing is known as demosaicing (or de-mosaicking). From the pioneering work of Bayer [1] to nowadays a lot of attention has been paid to the demosaicing topic (see [2] for a review). The use of a CFA, and the corresponding demosaicing process produce undesirable artifacts, such as color fringe, that are difficult to avoid.

Over the last two decades research has been devoted to the problem of reconstructing a high-resolution image from multiple undersampled, shifted, degraded frames with subpixel displacement errors (see [3] for a recent review). In our previous work [4][5] we addressed the high resolution problem from complete and also from incomplete observations within the general framework of frequency

---

\* This work has been supported by the “Comisión Nacional de Ciencia y Tecnología” under contract TIC2003-00880.

domain multi-channel signal processing developed in [6]. In this paper we formulate the demosaicing problem as a high resolution problem from incomplete observations and therefore we propose a new way to looking at the problem of demosaicing.

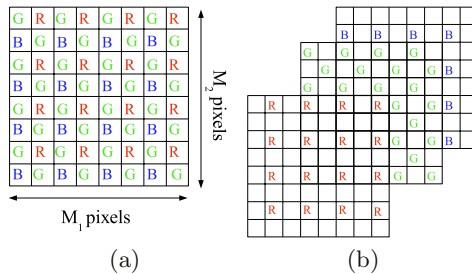
The rest of the paper is organized as follows. The problem formulation is described in section 2. In section 3 we describe the process for reconstructing each band of the color image and then examine how to iteratively estimate the high resolution color image. Experimental results are described in section 4. Finally, section 5 concludes the paper.

## 2 Problem Formulation

Consider a Bayer camera with a Color Filter Array (CFA) over one CCD with  $M_1 \times M_2$  pixels, as shown in Fig. 1(a). Assuming that the camera has three  $M_1 \times M_2$  CCDs (as is usually the case), one for each  $R, G, B$  channels, the observed image is given by

$$\mathbf{g} = (\mathbf{g}^{Rt}, \mathbf{g}^{Gt}, \mathbf{g}^{Bt})^t, \tag{1}$$

where  $t$  denotes the transpose of a vector or a matrix and each one of the  $M_1 \times M_2$  column vectors  $\mathbf{g}^c$ ,  $c \in \{R, G, B\}$ , results from the lexicographic ordering of the two-dimensional signal in the  $R, G$  and  $B$  channels, respectively.



**Fig. 1.** (a) Pattern of channel observations for a Bayer camera with CFA; (b) Observed low resolution channels (the array in (a) and all the arrays in (b) are of the same size)

Due to the presence of the CFA we do not observe  $\mathbf{g}$  but an *incomplete* subset of it, as shown in Fig. 1(b). Let us characterize these observed values in the Bayer camera. Let  $N_1 = M_1/2$  and  $N_2 = M_2/2$ ; then the 1-D downsampling matrices  $\mathbf{D}_l^x$  and  $\mathbf{D}_l^y$  are defined by

$$\mathbf{D}_l^x = \mathbf{I}_{N_1} \otimes \mathbf{e}_l^t, \quad \mathbf{D}_l^y = \mathbf{I}_{N_2} \otimes \mathbf{e}_l^t, \tag{2}$$

where  $\mathbf{I}_{N_i}$  is the  $N_i \times N_i$  identity matrix,  $\mathbf{e}_l$  is the  $2 \times 1$  unit vector whose nonzero element is in the  $l$ -th position,  $l \in \{0, 1\}$  and  $\otimes$  denotes the Kronecker product

operator. The  $(N_1 \times N_2) \times (M_1 \times M_2)$  2D downsampling matrix is now given by  $\mathbf{D}_{l1,l2} = \mathbf{D}_{l1}^x \otimes \mathbf{D}_{l2}^y$ , with  $l1, l2 \in \{0, 1\}$ .

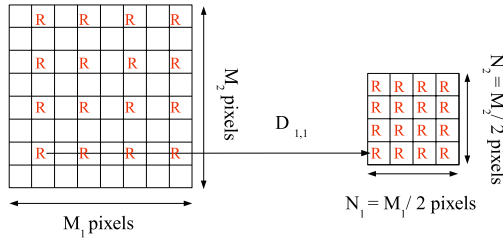
Using the above downsampling matrices, the subimage of  $\mathbf{g}$  which has been observed,  $\mathbf{g}^{\text{obs}}$ , may be viewed as the incomplete set of  $N_1 \times N_2$  low resolution images

$$\mathbf{g}^{\text{obs}} = (\mathbf{g}_{1,1}^{Rt}, \mathbf{g}_{1,0}^{Gt}, \mathbf{g}_{0,1}^{Gt}, \mathbf{g}_{0,0}^{Bt})^t, \quad (3)$$

where

$$\mathbf{g}_{1,1}^R = \mathbf{D}_{1,1}\mathbf{g}^R, \quad \mathbf{g}_{1,0}^G = \mathbf{D}_{1,0}\mathbf{g}^G, \quad \mathbf{g}_{0,1}^G = \mathbf{D}_{0,1}\mathbf{g}^G, \quad \mathbf{g}_{0,0}^B = \mathbf{D}_{0,0}\mathbf{g}^B. \quad (4)$$

As an example Fig. 2 illustrates how  $\mathbf{g}_{1,1}^R$  is obtained. Note that the origin of coordinates is located in the bottom-left side of the array. We have one observed  $N_1 \times N_2$  low resolution image at  $R$ , two at  $G$  and one at  $B$  channels.



**Fig. 2.** Process to obtain the low resolution observed  $R$  channel

We now assume that  $\mathbf{g}$  in equation (1) can be written as

$$\mathbf{g} = \begin{pmatrix} \mathbf{g}^R \\ \mathbf{g}^G \\ \mathbf{g}^B \end{pmatrix} = \begin{pmatrix} \mathbf{f}^R \\ \mathbf{f}^G \\ \mathbf{f}^B \end{pmatrix} + \begin{pmatrix} \mathbf{n}^R \\ \mathbf{n}^G \\ \mathbf{n}^B \end{pmatrix} = \mathbf{f} + \mathbf{n} \quad (5)$$

where  $\mathbf{f}$  denotes the real high resolution color image we are trying to estimate and  $\mathbf{n}$  denotes white independent uncorrelated noise between and within channels with variance  $1/\beta^c$  in channel  $c \in \{R, G, B\}$ . Substituting this equation in equation (4) we have that the discrete low-resolution observed images can be written as

$$\begin{aligned} \mathbf{g}_{1,1}^R &= \mathbf{D}_{1,1}\mathbf{f}^R + \mathbf{D}_{1,1}\mathbf{n}^R, & \mathbf{g}_{1,0}^G &= \mathbf{D}_{1,0}\mathbf{f}^G + \mathbf{D}_{1,0}\mathbf{n}^G, \\ \mathbf{g}_{0,1}^G &= \mathbf{D}_{0,1}\mathbf{f}^G + \mathbf{D}_{0,1}\mathbf{n}^G, & \mathbf{g}_{0,0}^B &= \mathbf{D}_{0,0}\mathbf{f}^B + \mathbf{D}_{0,0}\mathbf{n}^R, \end{aligned} \quad (6)$$

where we have the following distributions for the subsampled noise

$$\begin{aligned} \mathbf{D}_{1,1}\mathbf{n}^R &\sim N(0, 1/\beta^R I_{N_1 \times N_2}), & \mathbf{D}_{1,0}\mathbf{n}^G &\sim N(0, 1/\beta^G I_{N_1 \times N_2}), \\ \mathbf{D}_{0,1}\mathbf{n}^G &\sim N(0, 1/\beta^G I_{N_1 \times N_2}), & \mathbf{D}_{0,0}\mathbf{n}^B &\sim N(0, 1/\beta^B I_{N_1 \times N_2}). \end{aligned} \quad (7)$$

From the above formulation, our goal has become the reconstruction of a complete *RGB*  $M_1 \times M_2$  high resolution image  $\mathbf{f}$  from the incomplete set of observations,  $\mathbf{g}^{\text{obs}}$  in equation (3). In other words, the demosaicing problem has taken the form now of a superresolution reconstruction one. We can therefore apply the theory developed in [5, 7], but taking into account that we are dealing with multichannel images and that the relationship between channels should also be taken into account [8].

### 3 Bayesian Reconstruction of the Color Image

Let us consider first the reconstruction of channel  $c$  assuming that the observed data  $\mathbf{g}^{\text{obs } c}$  and also the real images  $\mathbf{f}^{c'}$  and  $\mathbf{f}^{c''}$ , with  $c' \neq c$  and  $c'' \neq c$ , are available.

In order to apply the Bayesian paradigm to this problem we define  $p_c(\mathbf{f}^c)$ ,  $p_c(\mathbf{f}^{c'}|\mathbf{f}^c)$ ,  $p_c(\mathbf{f}^{c''}|\mathbf{f}^c)$ , and  $p_c(\mathbf{g}^{\text{obs } c}|\mathbf{f}^c)$  and use the global distribution

$$p_c(\mathbf{f}^c, \mathbf{f}^{c'}, \mathbf{f}^{c''}, \mathbf{g}^{\text{obs } c}) = p_c(\mathbf{f}^c)p_c(\mathbf{f}^{c'}|\mathbf{f}^c)p_c(\mathbf{f}^{c''}|\mathbf{f}^c)p_c(\mathbf{g}^{\text{obs } c}|\mathbf{f}^c). \quad (8)$$

Smoothness within channel  $c$  is modelled by the introduction of the following prior distribution for  $\mathbf{f}^c$

$$p(\mathbf{f}^c|\alpha^c) \propto (\alpha^c)^{M_1 \times M_2/2} \exp \left[ -\frac{1}{2} \alpha^c \|\mathbf{C}\mathbf{f}^c\|^2 \right], \quad (9)$$

where  $\alpha^c > 0$  and  $\mathbf{C}$  denotes the Laplacian operator.

To define  $p_c(\mathbf{f}^{c'}|\mathbf{f}^c)$  and similarly  $p_c(\mathbf{f}^{c''}|\mathbf{f}^c)$ , we proceed as follows. A two-level bank of undecimated separable two-dimensional filters constructed from a low-pass filter  $H_l$  (with impulse response  $h_l = [1 \ 2 \ 1]/4$ ) and a high-pass filter  $H_h$  ( $h_h = [1 \ -2 \ 1]/4$ ) is applied to  $\mathbf{f}^{c'} - \mathbf{f}^c$  obtaining the approximation subband  $W_{ll}(\mathbf{f}^{c'} - \mathbf{f}^c)$ , and the horizontal  $W_{lh}(\mathbf{f}^{c'} - \mathbf{f}^c)$ , vertical  $W_{hl}(\mathbf{f}^{c'} - \mathbf{f}^c)$  and diagonal  $W_{hh}(\mathbf{f}^{c'} - \mathbf{f}^c)$  detail subbands [9] (see Fig. 3); where

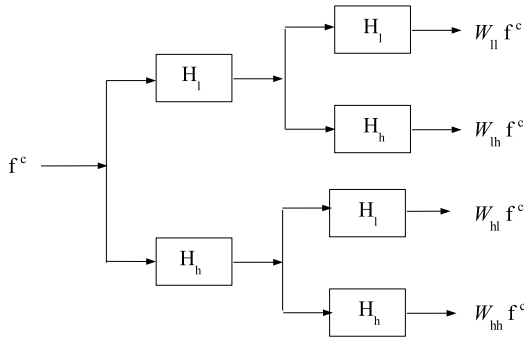
$$W_{uv} = H_u \otimes H_v, \text{ for } uv \in \{ll, lh, hl, hh\}. \quad (10)$$

With this decomposition differences between channels, for high frequency components, are penalized by the introduction of the following probability distribution

$$p_c(\mathbf{f}^{c'}|\mathbf{f}^c, \gamma^{cc'}) \propto |A(\gamma^{cc'})|^{-1/2} \exp \left[ -\frac{1}{2} \sum_{uv \in \mathcal{HB}} \gamma_{uv}^{cc'} \|W_{uv}(\mathbf{f}^{c'} - \mathbf{f}^c)\|^2 \right], \quad (11)$$

where  $\mathcal{HB} = \{lh, hl, hh\}$ ,  $\gamma_{\mu\nu}^{cc'}$  measures the similarity of the  $uv$  band of the  $c$  and  $c'$  channels,  $\gamma^{cc'} = \{\gamma_{uv}^{cc'}|uv \in \mathcal{HB}\}$ , and

$$A(\gamma^{cc'}) = \sum_{uv \in \mathcal{HB}} \gamma_{uv}^{cc'} W_{uv}^t W_{uv}. \quad (12)$$



**Fig. 3.** Two-level filter bank

From the model in Eq. (6), we have

$$p_c(\mathbf{g}^{\text{obs } c} | \mathbf{f}^c, \beta^c) \propto \begin{cases} \beta^R{}^{N_1 \times N_2 / 2} \exp \left[ -\frac{\beta^R}{2} \|\mathbf{g}_{1,1}^R - \mathbf{D}_{1,1} \mathbf{f}^R\|^2 \right] & \text{if } c = R \\ \beta^G{}^{N_1 \times N_2} \exp \left[ -\frac{\beta^G}{2} (\|\mathbf{g}_{1,0}^G - \mathbf{D}_{1,0} \mathbf{f}^G\|^2 + \|\mathbf{g}_{0,1}^G - \mathbf{D}_{0,1} \mathbf{f}^G\|^2) \right] & \text{if } c = G \\ \beta^B{}^{N_1 \times N_2 / 2} \exp \left[ -\frac{\beta^B}{2} \|\mathbf{g}_{0,0}^B - \mathbf{D}_{0,0} \mathbf{f}^B\|^2 \right] & \text{if } c = B \end{cases} \quad (13)$$

Note that from the above definitions of the probability density functions, the distribution in equation (8) depends on a set of unknown parameters and has to be properly written as

$$p_c(\mathbf{f}^c, \mathbf{f}^{c'}, \mathbf{f}^{c''}, \mathbf{g}^{\text{obs } c} | \Theta^c) \quad (14)$$

where

$$\Theta^c = (\alpha_c, \gamma^{cc'}, \gamma^{cc''}, \beta^c). \quad (15)$$

Having defined the involved distributions and the unknown parameters, the Bayesian analysis is performed to estimate the parameter vector  $\Theta^c$  and the unknown high resolution band  $\mathbf{f}^c$ . It is important to remember that we are assuming that  $\mathbf{f}^{c'}$  and  $\mathbf{f}^{c''}$  are known.

The process to estimate  $\Theta^c$  and  $\mathbf{f}^c$  is described by the following algorithm which corresponds to the so called *evidence analysis* within the Bayesian paradigm [10].

**Algorithm 1** (Estimation of  $\Theta^c$  and  $\mathbf{f}^c$  assuming that  $\mathbf{f}^{c'}$  and  $\mathbf{f}^{c''}$  are known)

Given  $\mathbf{f}^{c'}$  and  $\mathbf{f}^{c''}$

1. Find

$$\hat{\Theta}^c(\mathbf{f}^{c'}, \mathbf{f}^{c''}) = \arg \max_{\Theta^c} p_c(\mathbf{f}^{c'}, \mathbf{f}^{c''}, \mathbf{g}^{\text{obs } c} | \Theta^c) \quad (16)$$

$$= \arg \max_{\Theta^c} \int_{\mathbf{f}^c} p_c(\mathbf{f}^c, \mathbf{f}^{c'}, \mathbf{f}^{c''}, \mathbf{g}^{\text{obs } c} | \Theta^c) d\mathbf{f}^c \quad (17)$$

2. Find an estimate of channel  $c$  using

$$\hat{\mathbf{f}}^c(\hat{\Theta}^c(\mathbf{f}^{c'}, \mathbf{f}^{c''})) = \arg \max_{\mathbf{f}^c} p_c(\mathbf{f}^c | \mathbf{f}^{c'}, \mathbf{f}^{c''}, \mathbf{g}^{obs,c}, \hat{\Theta}^c(\mathbf{f}^{c'}, \mathbf{f}^{c''})) \quad (18)$$

Let us now assume that we have initial estimates of the three channels,  $\mathbf{f}^R(0)$ ,  $\mathbf{f}^G(0)$  and  $\mathbf{f}^B(0)$ ; then we can improve the quality of the reconstruction by using the following procedure

**Algorithm 2** (Reconstruction of the color image)

1. Given  $\mathbf{f}^R(0)$ ,  $\mathbf{f}^G(0)$  and  $\mathbf{f}^B(0)$ , initial estimates of the bands of the color image and  $\Theta^R(0)$ ,  $\Theta^G(0)$  and  $\Theta^B(0)$  of the model parameters
2. Set  $k=0$
3. Calculate

$$\mathbf{f}^R(k+1) = \hat{\mathbf{f}}^R(\hat{\Theta}^R(\mathbf{f}^G(k), \mathbf{f}^B(k))) \quad (19)$$

by running Algorithm 1 on channel  $R$  with  $\mathbf{f}^G = \mathbf{f}^G(k)$  and  $\mathbf{f}^B = \mathbf{f}^B(k)$

4. Calculate

$$\mathbf{f}^G(k+1) = \hat{\mathbf{f}}^G(\hat{\Theta}^G(\mathbf{f}^R(k+1), \mathbf{f}^B(k))) \quad (20)$$

by running Algorithm 1 on channel  $G$  with  $\mathbf{f}^R = \mathbf{f}^R(k+1)$  and  $\mathbf{f}^B = \mathbf{f}^B(k)$

5. Calculate

$$\mathbf{f}^B(k+1) = \hat{\mathbf{f}}^B(\hat{\Theta}^B(\mathbf{f}^R(k+1), \mathbf{f}^G(k+1))) \quad (21)$$

by running Algorithm 1 on channel  $B$  with  $\mathbf{f}^R = \mathbf{f}^R(k+1)$  and  $\mathbf{f}^G = \mathbf{f}^G(k+1)$

6. Set  $k=k+1$  and go to step 3 until a convergence criterion is met.

## 4 Experimental Results

A number of simulations have been performed with the proposed algorithm. Figure 4 shows a subset of images of size  $256 \times 384$ , taken from [11], used in the experiments. These images were sampled applying a Bayer pattern to get the observed images that are to be reconstructed.

The proposed Algorithm 2 was run using as initial image estimates bilinearly interpolated images, and the values  $\alpha^{c(0)} = 0.01$ ,  $\beta^{c(0)} = 1000.0$  and  $\gamma_{uv}^{cc'(0)} = 2.0$  (for all  $uv \in \mathcal{HB}$  and  $c' \neq c$ ) for all  $c \in \{R, G, B\}$ . The convergence criterion utilized was  $\frac{\|\mathbf{f}^c(k+1) - \mathbf{f}^c(k)\|^2}{\|\mathbf{f}^c(k)\|^2} \leq 10^{-7}$ .

Table 1 compares the results obtained by bilinear interpolation, the methods proposed by Laroche [12], Kimmel [11], Gunturk [9] and Algorithm 2. Their performance were evaluated by measuring the SNR improvement in dB, given by

$$\Delta_{SNR} = 10 \times \log_{10} \left[ \frac{\|\mathbf{f}^c - \mathbf{g}^{pad,c}\|^2}{\|\mathbf{f}^c - \hat{\mathbf{f}}^c\|^2} \right],$$

for  $c \in \{R, G, B\}$ , where  $\mathbf{f}^c$  and  $\hat{\mathbf{f}}^c$  are the original and estimated high resolution images, and  $\mathbf{g}^{pad,c}$  is the result of padding missing values at  $\mathbf{g}^{obs,c}$  (equation 3) with zeroes. Figure 5 shows an enlargement of a small region of the restorations of the image in Figure 4(c) by the different methods under comparison.

Finally, Algorithm 1 takes 5 secs of CPU time on a Intel Xeon processor at 3.2 GHz, with 4 GB of RAM, for images of the considered size.

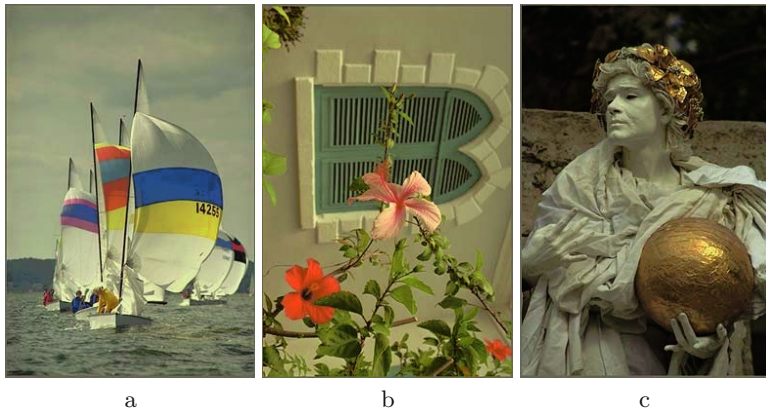


Fig. 4. Images used in the experiments

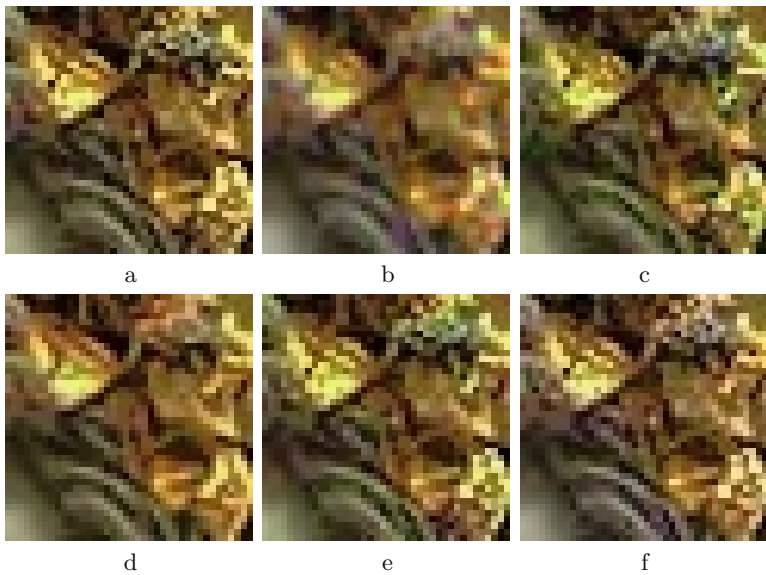


Fig. 5. Details of the (a) original image , (b) bilinear reconstruction, (c) Methods of Laroche [12], (d) Kimmel [11], (e) Gunturk [9] and (f) our method

Table 1.  $\Delta_{SNR}$  (dB)

image	bilinear			Laroche [12]			Kimmel [11]			Gunturk [9]			our method		
	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
Fig. 4.(a)	21.0	24.0	20.8	23.1	26.6	27.1	28.2	29.5	28.3	30.3	32.7	29.0	28.4	31.0	29.0
Fig. 4.(b)	20.3	21.7	17.6	18.5	23.9	21.9	26.2	25.5	23.0	27.8	29.5	25.9	29.0	28.6	23.6
Fig. 4.(c)	18.9	19.9	17.6	20.9	21.2	22.2	25.3	24.6	22.6	28.4	30.2	23.6	27.2	28.8	26.7



## 5 Conclusions

In this paper the color demosaicing problem has been formulated from a super-resolution point of view. A new method for estimating both the reconstructed color images and the model parameters, within the Bayesian framework, was obtained. Based on the presented experimental results, the new method outperforms bilinear interpolation and the methods in [11] and [12], while it performs comparably to the method in [9].

## References

1. Bayer, B.E.: Color imaging array (1976) United State Patent 3,971,065.
2. Ramanath, R.: Interpolation Methods for the Bayer Color Array. PhD thesis, North Carolina State University (2000)
3. Alvarez, L., Molina, R., Katsaggelos, A.: High resolution images from a sequence of low resolution observations. In Reed, T.R., ed.: *Digital Image Sequence Processing, Compression and Analysis*. CRC Press (2004) 233–259
4. Mateos, J., Molina, R., Katsaggelos, A.: Bayesian high resolution image reconstruction with incomplete multisensor low resolution systems. In: *2003 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP2003)*. Volume III., Hong Kong (2003) 705–708
5. Molina, R., Vega, M., Abad, J., Katsaggelos, A.: Parameter estimation in Bayesian high-resolution image reconstruction with multisensors. *IEEE Transactions on Image Processing* **12** (2003) 1655–1667
6. Katsaggelos, A.K., Lay, K.T., Galatsanos, N.P.: A general framework for frequency domain multi-channel signal processing. *IEEE Trans. Image Processing*, **2** (1993) 417–420
7. Mateos, J., Vega, M., Molina, R., Katsaggelos, A.: Bayesian image estimation from an incomplete set of blurred, undersampled low resolution images. In: *1st Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA2003)*, Lecture Notes in Computer Science. Volume 2652. (2003) 445–452
8. Molina, R., Mateos, J., Katsaggelos, A., Vega, M.: Bayesian multichannel image restoration using compound Gauss-Markov random fields. *IEEE Transactions on Image Processing* **12** (2003) 1642–1654
9. Gunturk, B.K., Altunbasak, Y., Mersereau, R.: Color plane interpolation using alternating projections. *IEEE Trans. Image Processing*, **11** (2002) 997–1013
10. Molina, R., Katsaggelos, A.K., Mateos, J.: Bayesian and regularization methods for hyperparameter estimation in image restoration. *IEEE Trans. Image Processing* **8** (1999) 231–246
11. Kimmel, R.: Demosaicing: Image reconstruction from color CCD samples. *IEEE Trans. on Image Processing* **8** (1999) 1221
12. Laroche, C.A., Prescott, M.A.: Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients (1994) United State Patent 5,373,322.

# A Fast and Exact Algorithm for Total Variation Minimization

Jérôme Darbon<sup>1,2</sup> and Marc Sigelle<sup>2</sup>

<sup>1</sup> EPITA Research and Development Laboratory (LRDE)  
14-16 rue Voltaire F-94276 Le Kremlin-Bicêtre, France

`jerome.darbon@{lrde.epita.fr, enst.fr}`

<sup>2</sup> ENST / LTCI CNRS UMR 5141  
46 rue Barrault, F-75013 Paris, France  
`marc.sigelle@enst.fr`

**Abstract.** This paper deals with the minimization of the total variation under a convex data fidelity term. We propose an algorithm which computes an exact minimizer of this problem. The method relies on the decomposition of an image into its level sets. Using these level sets, we map the problem into optimizations of independent binary Markov Random Fields. Binary solutions are found thanks to graph-cut techniques and we show how to derive a fast algorithm. We also study the special case when the fidelity term is the  $L^1$ -norm. Finally we provide some experiments.

## 1 Introduction

Minimization of the total variation (tv) for image reconstruction is of great importance for image processing applications [1, 17, 19, 21, 22]. It has been shown that these minimizers live in the space of bounded variation [9] which preserves edges and allows for sharp boundaries. In this paper we propose a new and fast algorithm which computes an exact solution of tv minimization-based problems.

Assume  $u$  is an image defined on  $\Omega$  then its total variation is  $tv(u) = \int_{\Omega} |\nabla u|$ , where the gradient is taken in the distributional sense. A classical approach to minimize tv is achieved by a gradient descent [24] which yields the following evolution equation  $\frac{\partial u}{\partial t} = \operatorname{div} \left( \frac{\nabla u}{|\nabla u| + \epsilon} \right)$ . To avoid division by zero,  $\epsilon$  is set to a small positive value. In [5], Chambolle reformulates tv minimization problem using duality. Using this formulation he proposes a fast algorithm. In [19], Pollak *et al.* present a fast algorithm which provide the exact solution in one dimension. However only an approximation is available in higher dimensions. After a discretization, tv minimization can be reformulated as a minimization problem involving a Markov Random Field (MRF). In [4], Boykov *et al.* present a fast approximation minimization algorithm based on graph cuts for MRF. An algorithm which computes an exact solution for MRF where the prior is convex is presented in [12]. It is also based on graph-cuts.

In this paper, we assume  $u$  and  $v$  are two images defined on  $\Omega$ . Thus we are interested in minimizing the following functional:

$$E(u) = \int_{\Omega} f(u(x), v(x)) dx + \beta \int_{\Omega} |\nabla u|. \quad (1)$$

We assume that the attachment to data term is a convex function of  $u(\cdot)$ , such as:  $f(u(x), v(x)) = |u(x) - v(x)|^p$  for the  $L^p$  case ( $p = 1, 2$ ), and that the regularization parameter  $\beta$  is some positive constant. In this paper, we propose a fast algorithm which computes an exact minimizer of problem 1. It relies on reformulating this problem into independent binary MRFs attached to each level set of an image. Exact minimization is performed thanks to a minimum cost cut algorithm.

The rest of this paper is organized as follows. In section 2 we map the original problem 1 into independent binary Markov Random Field optimizations. In section 3, a fast algorithm based on graph cuts is presented. In section 4 we shed new lights on tv minimization under the  $L^1$ -norm as fidelity term. Finally we draw some conclusions in section 5.

## 2 Formulation Using Level Sets and MRF

For the rest of this paper we assume that  $u$  takes values in the discrete set  $[0, L - 1]$  and is defined on a discrete lattice  $S$ . We denote by  $u_s$  the value of the image  $u$  at the site  $s \in S$ . Let us decompose an image into its level sets using the decomposition principle [11]. It corresponds to considering the thresholding image  $u^\lambda$  where  $u_s^\lambda = \mathbb{1}_{u_s \leq \lambda}$ . One can reconstruct the original image from its level sets using  $u_s = \min\{\lambda, u_s^\lambda = 1\}$ .

### 2.1 Reformulation into Binary MRFs

The coarea formula states that for any function  $u$  which belongs to the space of bounded variation [9] one has  $tv(u) = \int_{\mathbb{R}} P(u^\lambda) d\lambda$  almost surely. In the

discrete case, we write  $tv(u) = \sum_{\lambda=0}^{L-2} P(u^\lambda)$ , where  $P(u^\lambda)$  is the perimeter of  $u^\lambda$

(notice that  $u_s^{L-1} = 1$  for every  $s \in S$ , which explains the previous summation up to  $L - 2$ .) Let us define the neighboring relationship between two sites  $s$  and  $t$  as  $s \sim t$ . The associated cliques of order two are noted as  $(s, t)$ . This enables to estimate the perimeter using the approach proposed in [14]. Thus we have

$tv(u) = \sum_{\lambda=0}^{L-2} \sum_{(s,t)} w_{st} |u_s^\lambda - u_t^\lambda|$ , where  $w_{st}$  is set to 0.26 and 0.19 for the four- and eight- connected neighborhood, respectively.

**Proposition 1** *The discrete version of the energy  $E(u)$  rewrites as*

$$E(u) = \sum_{\lambda=0}^{L-2} E^\lambda(u^\lambda) + C \quad , \quad \text{where} \quad (2)$$

$$E^\lambda(u^\lambda) = \beta \left[ \sum_{(s,t)} w_{st} ((1 - 2u_t^\lambda) u_s^\lambda + u_t^\lambda) \right] + \sum_{s \in \Omega} (g_s(\lambda + 1) - g_s(\lambda))(1 - u_s^\lambda) \quad (3)$$

$$g_s(x) = f(x, v_s) \quad \forall s \in S \quad \text{and} \quad C = \sum_{s \in \Omega} g_s(0)$$

**Proof:** Using the following property for binary variables  $a, b$ :  $|a - b| = a + b - 2ab$ , and starting from the previous equality obtained from the coarea formula we have

$$tv(u) = \sum_{\lambda=0}^{L-2} \sum_{(s,t)} w_{st} ((1 - 2u_t^\lambda) u_s^\lambda + u_t^\lambda) \quad .$$

Moreover the following decomposition property holds for any function  $g$ :

$$\begin{aligned} \forall k \in [0, L - 1] \quad g(k) &= \sum_{\lambda=0}^{k-1} ((g(\lambda + 1) - g(\lambda)) + g(0)) \\ &= \sum_{\lambda=0}^{L-2} (g(\lambda + 1) - g(\lambda)) \mathbb{1}_{\lambda < k} + g(0) \end{aligned}$$

(note that this formula is coherent for both  $k = 0$  and  $k = L - 1$ ). Thus, by defining  $g_s(u_s) = f(u_s, v_s)$  and since  $\mathbb{1}_{\lambda < u_s} = 1 - u_s^\lambda$ , we have

$$f(u_s, v_s) = g_s(u_s) = \sum_{\lambda=0}^{L-2} (g_s(\lambda + 1) - g_s(\lambda)) (1 - u_s^\lambda) + g_s(0) \quad .$$

This concludes the proof.  $\square$

Note that each  $E^\lambda(u^\lambda)$  is a binary MRF with an Ising prior model. To minimize  $E(\cdot)$  one can minimize all  $E^\lambda(\cdot)$  independently. Thus we get a family  $\{\hat{u}^\lambda\}$  which are respectively minimizers of  $E^\lambda(\cdot)$ . Clearly the summation will be minimized and thus we have a minimizer of  $E(\cdot)$  provided this family is monotone:

$$\hat{u}^\lambda \leq \hat{u}^\mu \quad \forall \lambda < \mu \quad . \quad (4)$$

If this property holds then the optimal solution is given by [11]:  $\hat{u}_s = \min\{\lambda, \hat{u}_s^\lambda = 1\} \forall s$ . If property 4 does not hold, then the family  $\{u^\lambda\}$  is not a function.

## 2.2 A Lemma Based on Coupled Markov Chains

Since the MRF posterior energy is decomposable into levels, it is useful to define the ‘‘local neighborhood configurations’’:  $N_s = \{u_t\}_{t \sim s}$  and  $N_s^\lambda = \{u_t^\lambda\}_{t \sim s} \forall \lambda \in [0, L - 2]$ . In [8] the following lemma was established:

**Lemma 1** *If the local conditional posterior energy at each site  $s$  writes as*

$$E(u_s \mid N_s, v_s) = \sum_{\lambda=0}^{L-2} ( \Delta\phi_s(\lambda) u_s^\lambda + \chi_s(\lambda) ) \tag{5}$$

where  $\Delta\phi_s(\lambda)$  is a non-increasing function of  $\lambda$  and  $\chi_s(\lambda)$  does not depend on  $u_s^\lambda$ , then one can exhibit a “coupled” stochastic algorithm minimizing each total posterior energy  $E^\lambda(u^\lambda)$  while preserving the monotone condition:  $\forall s, u_s^\lambda \nearrow$  with  $\lambda$ .

In other words, given a binary solution  $u^*$  to the problem  $E^k$ , there exists at least one solution  $\hat{u}$  to the problem  $E^l$  such that  $u^* \leq \hat{u} \forall k \leq l$ . The proof of the Lemma relies on coupled Markov chains [20].

**Proof:** We endow the space of binary configurations by the following order:  $u \leq v$  iff  $u_s \leq v_s \forall s \in \Omega$ . From the decomposition (5) the local conditional posterior energy at level value  $\lambda$  is  $E(u_s^\lambda \mid N_s^\lambda, v_s) = \Delta\phi_s(\lambda) u_s^\lambda + \chi_s(\lambda)$ . Thus let us define the following Gibbs local conditional posterior probability:

$$P_s(\lambda) = P(u_s^\lambda = 1 \mid N_s^\lambda, v_s) = \frac{\exp -\Delta\phi_s(\lambda)}{1 + \exp -\Delta\phi_s(\lambda)} = \frac{1}{1 + \exp \Delta\phi_s(\lambda)}. \tag{6}$$

With the conditions of the Lemma 1, this latter expression is clearly a monotone non-decreasing function of  $\lambda$ .

Let us now design a “coupled” Gibbs sampler for the  $L - 1$  binary images in the following sense: first consider a visiting order of the sites (tour). When a site  $s$  is visited, pick up a *single* random number  $\rho_s$  uniformly distributed in  $[0, 1]$ . Then, for each value of  $\lambda$ , assign:  $u_s^\lambda = 1$  if  $0 \leq \rho_s \leq P_s(\lambda)$  or else  $u_s^\lambda = 0$  (this is the usual way to draw a binary value according to its probability, except that we use here the same random number  $\rho_s$  for all the  $L - 1$  binary images.) From the non-decreasing monotony of (6) it is seen that the set of assigned binary values at site  $s$  satisfies  $u_s^\lambda = 1 \Rightarrow u_s^\mu = 1 \forall \mu > \lambda$ . The monotone property  $u^\lambda \leq u^\mu \forall \lambda < \mu$  is thus preserved. Clearly, this property also extends to a series of  $L - 1$  coupled Gibbs samplers having *the same* positive temperature  $T$  when visiting a given site  $s$ : it suffices to replace  $\Delta\phi_s(\lambda)$  by  $\Delta\phi_s(\lambda) / T$  in (6). Hence, this property also holds for a series of  $L - 1$  coupled Simulated Annealing algorithms [10] where a *single* temperature  $T$  boils down to 0 (either after each visited site  $s$  or at the beginning of each tour [25].) □

It must be noticed that our Lemma gives a *sufficient* condition for the simultaneous, “level-by-level independent” minimization of posterior energies while preserving the monotone property. We shall now prove the following property:

**Lemma 2** *The requirements stated by Lemma 1 are equivalent to these: all conditional energies  $E(u_s \mid N_s, v_s)$  are convex functions of grey level  $u_s \in [0, L - 1]$ , for any neighborhood configuration and local observed data.*

**Proof:** Since from (2) the total energy is “decomposable” on the levels, so are

the local conditional energies:  $E(u_s \mid N_s, v_s) = \sum_{\lambda=0}^{L-2} E^\lambda(u_s^\lambda \mid N_s^\lambda, v_s).$

Besides, since the local conditional posterior energy at level  $\lambda$  is a function of binary variable  $u_s^\lambda$ , it satisfies:

$$\begin{aligned}
 & E^\lambda(u_s^\lambda | N_s^\lambda, v_s) - E^\lambda(u_s^\lambda = 0 | N_s^\lambda, v_s) \\
 &= (E^\lambda(u_s^\lambda = 1 | N_s^\lambda, v_s) - E^\lambda(u_s^\lambda = 0 | N_s^\lambda, v_s)) u_s^\lambda
 \end{aligned}$$

which yields by identification with (5):

$$\Delta\phi_s(\lambda) = E^\lambda(u_s^\lambda = 1 | N_s^\lambda, v_s) - E^\lambda(u_s^\lambda = 0 | N_s^\lambda, v_s)$$

Now, in the transition  $\lambda \rightarrow \lambda + 1$ , only the following level variable does change:  $u_s^\lambda = 1 \rightarrow u_s^\lambda = 0$ . From the decomposition of conditional energies on levels, this means that only the level component  $E^\lambda(u_s^\lambda | N_s^\lambda, v_s)$  does change and thus:

$$\begin{aligned}
 & E(\lambda + 1 | N_s, v_s) - E(\lambda | N_s, v_s) \\
 &= E^\lambda(u_s^\lambda = 0 | N_s^\lambda, v_s) - E^\lambda(u_s^\lambda = 1 | N_s^\lambda, v_s) \\
 &= -\Delta\phi_s(\lambda)
 \end{aligned}$$

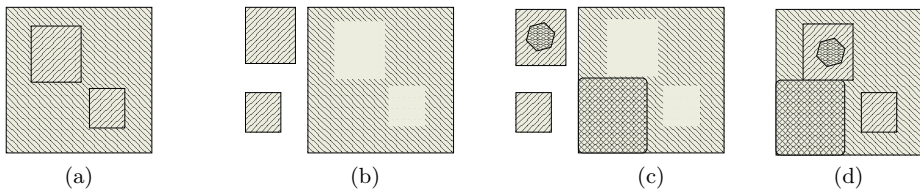
The monotone non-increasing condition on  $\Delta\phi_s(\lambda)$  is thus equivalent to:

$E(\lambda + 1 | N_s, v_s) - E(\lambda | N_s, v_s)$  is a *non-decreasing* function on  $[0, L - 2]$ .  $\square$   
 Clearly both  $L^1 + \text{TV}$  and  $L^2 + \text{TV}$  models enjoy this convexity property and satisfy thus the conditions of application of Lemma 1.

### 3 Minimization Algorithm

Although the previous section proves that the monotone property holds, it does not provide an algorithm to compute a solution. Our algorithm makes use of the formulation shown in equation 2 which allows independent optimizations. A natural algorithm, presented in [8], is to optimize independently each MRF. This leads to an algorithm which performs  $L - 1$  optimizations on binary images whose sizes are the same as the original image.

However, one can both drastically save computations using a divide and conquer approach. Such an approach requires to decompose a problem into smaller ones, then to solve these sub-problems and to recombine the sub-solutions to get an optimal solution. Our algorithm takes benefit of the following. Suppose we minimize at some level  $\lambda$ . Then, for all pixels of the minimizer we know whether they are below or above  $\lambda$ . Thus it is useless to take into account pixels above  $\lambda$  for further optimizations which only allow pixels to be lower than  $\lambda$ . Obviously, the same holds for pixels which are below  $\lambda$ . Then, every connected component (it defines a partition of the image) of the minimizer can be optimized independently from each others. The latter corresponds to the decomposition of the problem into subproblems. Once minimizers of subproblems are computed, they are recombined to yield an optimal solution. The recombination is straightforward since the decomposition was a partition. This process is depicted in Figure 1. A good choice to choose the threshold level  $\lambda$  is to use



**Fig. 1.** Illustration of our algorithm. The partition of the image after a minimization with respect to some level  $\lambda$  is shown on (a). The connected components of the image (a) are shown on (b); it corresponds to the decomposition of the problem into subproblems. Each subproblem is solved independently and the result is depicted in (c). Finally solutions of subproblems are recombined to yield the image (d).

**Table 1.** Time results (in seconds) with  $L^1$  data fidelity term for different weighted term  $\beta$ . Two time results are presented: time for our algorithm and time for the algorithm presented in [8] inside parentheses.

Image	$\beta = 1$	$\beta = 2$	$\beta = 3$
Lena (256x256)	0.37(7.31)	0.54(14.52)	0.72(16.41)
Lena (512x512)	1.56(31.10)	2.24(53.36)	2.84(101.33)
Woman (522x232)	0.53(16.03)	0.77(20.34)	1.03(23.86)

a dichotomic process. For instance, suppose the minimizer is a constant image, then our algorithm requires exactly  $\log_2(L)$  (we suppose  $L$  is a power of two) binary optimizations to compute it. This is in contrast compared to the  $L - 1$  required binary optimizations of the algorithm proposed in [8].

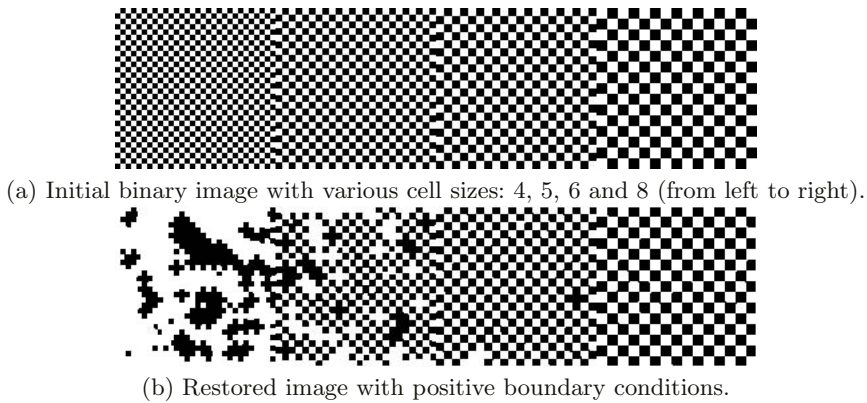
Optimization of a binary MRF can be performed exactly and efficiently using graph-cut techniques. It consists of building a graph such that its minimum cut gives an optimal labelling. We build the graph as proposed in [13]. Our implementation uses the minimum cut algorithm described in [3]. Time results (on a 3GHz Pentium IV) for our algorithm and the one presented in [8] are given in Table 1 for  $L^1$  fidelity. Note how our algorithm outperforms the other one.

## 4 Total Variation with $L^1$ Data Fidelity

The use of total variation with  $L^1$  data fidelity has already been studied in [2, 6, 15, 16]. However, the following is new as far as we know. Note that the Ising model fulfills the necessary condition provided that the interaction is attractive (i.e.  $\beta$  is non-negative) which is the case in our problems.

As a matter of fact, due to the equivalence of the Potts framework, the initial  $L_1 + TV$  restoration model (assign  $g_s(u_s) = |u_s - v_s| = \sum_{\lambda=0}^{L-2} |u_s^\lambda - v_s^\lambda|$  in (3)) is equivalent to an Ising model with constant magnetic field amplitude  $B = 1/2$  and constant interaction coefficient  $J = \beta/2$  over the whole range of levels.

It was shown, first semi-empirically [23] and then rigorously [18] that the 4-connected chessboard model exhibits a phase transition property. Namely if the basic cell size  $A$  satisfies:  $A \leq 4J/B = 4\beta$  then two ground states occur,



**Fig. 2.** Minimal energy configurations obtained by Simulated Annealing. Initial temperature  $T_0 = 16$  with decreasing step  $= 0.98$ ,  $\beta = 1.5$  (4-connectivity).



**Fig. 3.** Minimizers of TV with  $L^1$  fidelity. From left to right: original image, then minimizers for  $\beta = 1$ ,  $\beta = 2.1$ ,  $\beta = 3$ . Finally, some level lines of the minimizers (in the same order). Only level lines multiples of 10 are displayed.

corresponding to uniform binary images. In the opposite case, the unique ground state is the initial chessboard itself. In other words, and put in a rather “inexact” way, objects whose characteristic size is greater than  $4\beta$  are conveniently restored, whereas smaller objects are lost in the “background”. This property holds on the whole range of levels for the  $L^1 + TV$  model (See Fig. 2).

Moreover, it was shown in [7] that the continuous approach to this problem generates extra grey levels outside the initial grey level range, which is obviously not the case here. It happens because of the  $\epsilon$  introduced in the numerical scheme to avoid division by zero. Figure 3 depicts some results on the image woman. Note how well the contrast is preserved and how level lines are simplified.

## 5 Conclusion

In this paper we have presented an algorithm which computes an exact solution for the minimization of the total variation under a convex constraint. The method



relies on the decomposition of the problem into binary ones using the level sets of an image. Moreover, this algorithm is quite fast. Comparison to other algorithms with respect to time complexity must be made. Extension of this method to other type of regularization is in progress.

## References

1. S. Alliney. An algorithm for the minimization of mixed  $l^1$  and  $l^2$  norms with application to bayesian estimation. *IEEE Signal Processing*, 42(3):618–627, 1994.
2. S. Alliney. A property of the minimum vectors of a regularizing functional defined by means of the absolute norm. *IEEE Signal Processing*, 45(4):913–917, 1997.
3. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE PAMI*, 26(9):1124–1137, 2004.
4. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE PAMI*, 23(11):1222–1239, 2001.
5. A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, 2004.
6. T.F. Chan and S. Esedoğlu. Aspect of total variation regularized  $l^1$  function approximation. Technical Report 7, UCLA, 2004.
7. T.F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models. Technical report, September 2004.
8. J. Darbon and M. Sigelle. Exact Optimization of Discrete Constrained Total Variation Minimization Problems. In LNCS series vol. 3322, editor, *Tenth International Workshop on Combinatorial Image Analysis*, pages 540–549, 2004.
9. L. Evans and R. Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, 1992.
10. S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE PAMI*, 6(6):721–741, 1984.
11. F. Guichard and J.M. Morel. Mathematical morphology ”almost everywhere”. In *Proceedings of ISMM*, pages 293–303. Csiro Publishing, April 2002.
12. H. Ishikawa. Exact optimization for Markov random fields with priors. *IEEE PAMI*, 25(10):1333–1336, November 2003.
13. V. Kolmogorov and R. Zabih. What energy can be minimized via graph cuts? *IEEE PAMI*, 26(2):147–159, 2004.
14. H.T. Nguyen, M. Worring, and R. van den Boomgaard. Watersnakes: Energy-driven watershed segmentation. *IEEE PAMI*, 23(3):330–342, 2003.
15. M. Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. *SIAM J. Num. Anal.*, 40(3):965–994, 2002.
16. M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20:99–120, 2004.
17. S. Osher, A. Solé, and L. Vese. Image decomposition and restoration using total variation minimization and the  $\mathbf{H}^{-1}$  norm. *J. Mult. Model. and Simul.*, 1(3), 2003.
18. E. Pechersky, A. Maruani, and M. Sigelle. On Gibbs Fields in Image Processing. *Markov Processes and Related Fields*, 1(3):419–442, 1995.
19. I. Pollak, A.S. Willsky, and Y. Huang. Nonlinear evolution equations as fast and exact solvers of estimation problems. *IEEE Signal Processing*, 53(2):484–498, 2005.
20. J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and statistical mechanics. *Random Structures and Algorithms*, 9(1):223–252, 1996.

21. L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D.*, 60:259–268, 1992.
22. K. Sauer and C. Bouman. Bayesian estimation of transmission tomograms using segmentation based optimization. *IEEE Nuclear Science*, 39(4):1144–1152, 1992.
23. M. Sigelle. *Champs de Markov en Traitement d’Images et Modèles de la Physique Statistique: Application à la Relaxation d’Images de Classification*. PhD thesis, ENST <http://www.tsi.enst.fr/sigelle/tsi-these.html>, 1993.
24. C. Vogel and M. Oman. Iterative method for total variation denoising. *SIAM J. Sci. Comput.*, 17:227–238, 1996.
25. G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Applications of mathematics. Springer-Verlag, 2003.

# Phase Unwrapping via Graph Cuts<sup>\*</sup>

José M. Bioucas-Dias<sup>1</sup> and Gonçalo Valadão<sup>2</sup>

<sup>1</sup> Instituto de Telecomunicações - Instituto Superior Técnico  
Av. Rovisco Pais, 1049-001 Lisboa, Portugal  
Phone: 351 21 8418466

`bioucas@lx.it.pt`  
<sup>2</sup> ICIST - Instituto Superior Técnico  
Av. Rovisco Pais, 1049-001 Lisboa, Portugal  
Phone: 351 21 8418336  
`gvm@civil.ist.utl.pt`

**Abstract.** This paper presents a new algorithm for recovering the absolute phase from modulo- $2\pi$  phase, the so-called phase unwrapping (PU) problem. PU arises as a key step in several imaging technologies, from which we emphasize interferometric SAR and SAS, where topography is inferred from absolute phase measurements between two (or more) antennas and the terrain itself. The adopted criterion is the minimization of the  $L^p$  norm of phase differences [1], [2], usually leading to computationally demanding algorithms. Our approach follows the idea introduced in [3] of an iterative binary optimization scheme, the novelty being the casting onto a graph max-flow/min-cut formulation, for which there exists efficient algorithms. That graph formulation is based on recent energy minimization results via graph-cuts [4]. Accordingly, we term this new algorithm PUMF (for phase unwrapping max-flow). A set of experimental results illustrates the effectiveness of PUMF.

## 1 Introduction

Phase is an important property of many classes of signals. For instance, interferometric SAR (InSAR) uses two or more antennas to measure the phase between the antennas and the terrain; the topography is then inferred from the difference between those phases [5]. In magnetic resonance imaging (MRI), phase is used, namely, to determine magnetic field deviation maps, which are used to correct echo-planar image geometric distortions [6]. In optical interferometry, phase measurements are used to detect objects shape, deformation, and vibration [7].

In all the examples above, in spite of phase being a crucial information, the acquisition system can only measure phase modulo- $2\pi$ , the so-called principal phase value, or wrapped phase. Formally, we have

$$\phi = \psi + 2k\pi, \tag{1}$$

---

<sup>\*</sup> This work was supported by the Fundação para a Ciência e Tecnologia, under the projects POSI/34071/CPS/2000 and PDCTE/CPS/49967/2004.

where  $\phi$  is the true phase value (the so-called absolute value),  $\psi$  is the measured (wrapped) modulo- $2\pi$  phase value, and  $k \in \mathbb{Z}$  an integer number of wavelengths [2].

Phase unwrapping (PU) is the process of recovering the absolute phase  $\phi$  from the wrapped phase  $\psi$ . This is, however, an ill-posed problem, if no further information is added. In fact, an assumption taken by most phase unwrapping algorithms is that the absolute value of phase differences between neighbouring pixels is less than  $\pi$ , the so-called Itoh condition [8]. If this assumption is not violated, the absolute phase can be easily determined, up to a constant. Itoh condition might be violated if the true phase surface is discontinuous, or if only a noisy version of the wrapped phase is available. In either cases, PU becomes a very difficult problem, to which much attention has been devoted [2], [3].

Phase unwrapping approaches belong to one of these following classes: path following [9], minimum  $L^p$  norm [1], Bayesian [10], and parametric modelling [11].

Path following algorithms apply line integration schemes over the wrapped phase image, and basically rely on the assumption that Itoh condition holds along the integration path. Techniques employed to handle these inconsistencies include the so-called *residues branch cuts* [9] and *quality maps* [2].

Minimum norm methods exploit the fact that the differences between absolute phases of neighbour pixels, are equal to the wrapped differences between correspondent wrapped phases, if Itoh condition is met. Thus, these methods try to find a phase solution  $\phi$  for which  $L^p$  norm of the difference between absolute phase differences and wrapped phase differences (so a second order difference) is minimized. This is, therefore, a global minimization in the sense that all the observed phases are used to compute a solution. With  $p = 2$  we have a least squares method [12]. A drawback of the  $L^2$  norm is that this criterion tends to smooth discontinuities, unless they are provided as binary weights.  $L^1$  norm performs better than  $L^2$  norm in what discontinuity preserving is concerned. Such a criterion has been solved by Flynn [13], using network programming. With  $0 \leq p < 1$  the ability of preserving discontinuities is further increased at stake, however, of highly complex algorithms.

The Bayesian approach relies on a data-observation mechanism model, as well as a prior knowledge of the phase to be modelled. For instance in [14], a non-linear optimal filtering is applied, while in [15] an InSAR observation model is considered, and is taken into account not only the image phase, but also the *backscattering coefficient* and *correlation factor* images, which are jointly recovered from InSAR image pairs.

Finally, parametric algorithms constrain the unwrapped phase to a parametric surface. Low order polynomial surfaces are used in [11]. Very often in real applications just one polynomial is not enough to describe accurately the complete surface. In such cases the image is partitioned and different parametric models are applied to each partition [11].

## 1.1 Proposed Approach

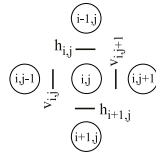
This paper proposes a new phase minimum  $L^p$  norm unwrapping algorithm, that minimizes the  $L^p$  norm of the complete set of phase differences between neigh-

bour pixels, with the additional constraint of being  $2\pi$ -congruent with wrapped phases.

The integer optimization problem we are led to is solved by a series of binary elementary optimizations as presented in [3] for the  $\mathbb{Z}\pi\text{M}$  algorithm. The present approach casts, however, the optimization problem as a max-flow/min-cut calculation on a certain graph, building on energy minimization results presented in [4]. The developed algorithm is valid for  $p \geq 1$ . Accordingly, we call the method to be presented, the PUMF algorithm (for PU-max-flow).

## 2 Problem Formulation

Adopting the representation used in [3], Fig.1 shows a pixel and its four neighbours along with the variables  $h$  and  $v$  signalling horizontal and vertical discontinuities respectively.



**Fig. 1.** Representation of the pixel  $(i,j)$  and its first order neighbours along with the variables  $h$  and  $v$  signalling horizontal and vertical discontinuities respectively.

The  $L^p$  norm of the difference between neighbouring pixels phases,  $2\pi$ -congruent with wrapped phases, is given by

$$E(\mathbf{k}|\boldsymbol{\psi}) = \sum_{ij \in \mathbb{Z}_1} |\Delta\phi_{ij}^h|^p \bar{v}_{ij} + |\Delta\phi_{ij}^v|^p \bar{h}_{ij}, \tag{2}$$

where  $(\cdot)^h$  and  $(\cdot)^v$  denotes pixel horizontal and vertical differences given by

$$\Delta\phi_{ij}^h = [2\pi(k_{ij} - k_{ij-1}) - \Delta\psi_{ij}^h], \quad k \in \mathbb{Z} \tag{3}$$

$$\Delta\phi_{ij}^v = [2\pi(k_{ij} - k_{i-1j}) - \Delta\psi_{ij}^v], \quad k \in \mathbb{Z} \tag{4}$$

$$\Delta\psi_{ij}^h = \psi_{ij-1} - \psi_{ij} \tag{5}$$

$$\Delta\psi_{ij}^v = \psi_{i-1j} - \psi_{ij}, \tag{6}$$

with  $p \geq 0$ ,  $\psi$  being the wrapped (observed) phase,  $\bar{h}_{ij} = 1 - h_{ij}$  and  $\bar{v}_{ij} = 1 - v_{ij}$  ( $h_{ij}, v_{ij} \in \{0, 1\}$ ) being binary horizontal and vertical discontinuities respectively, and  $(i, j) \in \mathbb{Z}_1$  where  $\mathbb{Z}_1 = \{(i, j) : i = 1, \dots, M, j = 1, \dots, N\}$ , and with  $M$  and  $N$  denoting the number of lines and columns respectively (i.e., the usual image pixel indexing 2D grid).

Our purpose is to find the integer image  $\mathbf{k}$  that minimizes energy (2),  $\mathbf{k}$  being such that  $\phi = 2\pi\mathbf{k} + \boldsymbol{\psi}$ , where  $\phi$  is the estimated unwrapped image;  $\mathbf{k}$  is the so-called *wrap-count* image. To achieve this goal, we compute a series of graph flow calculations for which efficient max-flow/min-cut algorithms exist.

### 3 Minimizing $E$ by a Sequence of Binary Optimizations

The following lemma, taken from [3], assures that if the minimum of  $E(\mathbf{k}|\boldsymbol{\psi})$  is not yet reached, then there exists a binary image  $\delta\mathbf{k}$  (i.e., the elements of  $\delta\mathbf{k}$  are all 0 or 1) such that  $E(\mathbf{k} + \delta\mathbf{k}|\boldsymbol{\psi}) < E(\mathbf{k}|\boldsymbol{\psi})$ .

**Lemma 1** *Let  $\mathbf{k}_1$  and  $\mathbf{k}_2$  be two wrap-count images such that*

$$E(\mathbf{k}_2|\boldsymbol{\psi}) < E(\mathbf{k}_1|\boldsymbol{\psi}). \quad (7)$$

*Then there exists a binary image  $\delta\mathbf{k}$  such that*

$$E(\mathbf{k}_1 + \delta\mathbf{k}|\boldsymbol{\psi}) < E(\mathbf{k}_1|\boldsymbol{\psi}). \quad (8)$$

*Proof.* The proof follows the same line of the one given in the appendix of [3] for  $p = 2$ , using the convexity of  $|x|^p$  with respect to  $x$ , for  $p \geq 1$ .

According to Lemma 1, we can iteratively compute  $\mathbf{k}^{t+1} = \mathbf{k}^t + \delta\mathbf{k}$ , where  $\delta\mathbf{k} \in \{0, 1\}^{MN}$  minimizes  $E(\mathbf{k}^t + \delta\mathbf{k}|\boldsymbol{\psi})$ , until the the minimum energy is reached.

#### 3.1 Mapping Binary Optimizations onto Graph Min-Cuts

Let  $k_{ij}^{t+1} = k_{ij}^t + \delta k_{ij}^t$  be the wrap-count at time  $t+1$  and pixel  $(i, j)$ . Introducing  $k_{ij}^{t+1}$  into (3) and (4), making some simple manipulations and introducing the obtained expressions into (2), we can rewrite energy  $E(\mathbf{k}|\boldsymbol{\psi})$  as a function of binary variables  $\delta k_{ij}^t$ , i.e.,

$$E(\mathbf{k}|\boldsymbol{\psi}) = \sum_{ij \in \mathbb{Z}_1} \underbrace{|2\pi(\delta k_{ij}^t - \delta k_{ij-1}^t) + a^h|^p \bar{v}_{ij}}_{E_h^{ij}(x_{ij-1}, x_{ij})} + \underbrace{|2\pi(\delta k_{ij}^t - \delta k_{i-1j}^t) + a^v|^p \bar{h}_{ij}}_{E_v^{ij}(x_{i-1j}, x_{ij})}, \quad (9)$$

where  $x_{ij} = \delta k_{ij}^t$ ,  $a^h = 2\pi(k_{ij}^t - k_{ij-1}^t) - \Delta\psi_{ij}^t$ , and  $a^v = 2\pi(k_{ij}^t - k_{i-1j}^t) - \Delta\psi_{ij}^t$ .

For simplicity, let us denote for a moment terms  $E_h^{ij}$  and  $E_v^{ij}$  by  $E^{ij}(x_k, x_l)$ . We have thus,  $E^{ij}(0, 0) = |a|^p \bar{d}_{ij}$ ,  $E^{ij}(1, 1) = |a|^p \bar{d}_{ij}$ ,  $E^{ij}(0, 1) = |2\pi + a|^p \bar{d}_{ij}$ , and  $E^{ij}(1, 0) = |-2\pi + a|^p \bar{d}_{ij}$ , where  $a$  represents  $a_h$  or  $a_v$  and  $\bar{d}_{ij}$  represents  $\bar{h}_{ij}$  or  $\bar{v}_{ij}$ .

So, we also have  $E^{ij}(0, 0) + E^{ij}(1, 1) = 2|a|^p \bar{d}_{ij}$ , and  $E^{ij}(0, 1) + E^{ij}(1, 0) = (|-2\pi + a|^p + |2\pi + a|^p) \bar{d}_{ij}$ . For  $p \geq 1$ , terms  $E(x_k, x_l)$  verify  $E^{ij}(0, 0) + E^{ij}(1, 1) \leq E^{ij}(0, 1) + E^{ij}(1, 0)$ , this following from the convexity of  $E(\mathbf{k}|\boldsymbol{\psi})$ .

We are now in conditions of using Theorem 4.1 stated in [4]:

**Theorem( $\mathcal{F}^2$  theorem) 1** *Let  $E$  be a function of  $n$  binary variables from the class  $\mathcal{F}^2$ , i.e.,*

$$E(x_1, \dots, x_n) = \sum_i E^i(x_i) + \sum_{i < j} E^{ij}(x_i, x_j). \quad (10)$$

*Then  $E$  is graph-representable if and only if each term  $E^{ij}$  satisfies the inequality*

$$E^{ij}(0, 0) + E(1, 1)^{ij} \leq E^{ij}(0, 1) + E^{ij}(1, 0). \quad (11)$$

*Proof.* See the proof in [4].

From the above theorem we can now state that energy  $E(\mathbf{k}|\psi)$  (2) is graph representable. In fact, it has the structure of function  $E$  in Theorem 1, with null one-variable terms. The inequality (11) was verified above.

The proof of the precedent theorem, presented in [4], shows how to construct that graph. First, build vertices and edges corresponding to each pair of neighbouring pixels, and then join these graphs together based on the additivity theorem also given in [4].

So, for each energy term  $E_h^{ij}$  and  $E_v^{ij}$  (see expression 9), we construct an ‘‘elementary’’ graph with four vertices  $\{s, t, v, v'\}$ , where  $\{s, t\}$  represents source and the sink, common to all terms, and  $\{v, v'\}$  represents the two pixels involved ( $v$  being the left (up) pixel and  $v'$  the right (down) pixel). Following very closely [4], we define a directed edge  $(v, v')$  with the weight  $E(0, 1) + E(1, 0) - E(0, 0) - E(1, 1)$ . Moreover, if  $E(1, 0) - E(0, 0) > 0$  we define an edge  $(s, v)$  with the weight  $E(1, 0) - E(0, 0)$  or, otherwise, we define an edge  $(v, t)$  with the weight  $E(0, 0) - E(1, 0)$ . In a similar way for vertex  $v'$ , if  $E(1, 1) - E(1, 0) > 0$  we define an edge  $(s, v')$  with weight  $E(1, 1) - E(1, 0) > 0$  or, otherwise, we define an edge  $(v', t)$  with the weight  $E(1, 0) - E(1, 1)$ .

In [4] it is shown that there is a one-to-one mapping between the configuration of  $(x_1, \dots, x_n)$ , and cuts leaving the source and the sink in disconnected components; furthermore, the cost of the cut is the value of the energy on that configuration. Therefore, minimizing the energy corresponds to computing the max-flow. As we have shown above, building on results from [3] and from [4], we can iteratively find an energy minimum through binary optimizations, based on max-flow calculation on a certain graph.

Algorithm 1 shows the pseudo-code for the Phase Unwrapping Max-Flow (PUMF) algorithm<sup>1</sup>.

## 4 Experimental Results

The results presented in this section were obtained by a MATLAB coding of the PUMF algorithm [max-flow was coded in C++ (see [16])].

Fig. 2(a) displays a noisy phase image to be unwrapped; it was synthesized from a Gaussian elevation height of  $14\pi$  rad, and standard deviations  $\sigma_i = 15$  and  $\sigma_j = 10$  pixels. This synthesis consists of generating a pair of SAR complex images, given the desired absolute phase surface and pair coherence [17]; this is done according to the InSAR observation model adopted in [3]. The wrapped phase image is then obtained, by computing the product of one image by the complex conjugate of the other, and finally taking the argument. The correlation coefficient,  $0 \leq \alpha \leq 1$ , of the associated InSAR pair is, in this case,  $\alpha = 0.8$ . This value is low enough to induce a large number of phase jumps, making the unwrapping a very difficult task. Fig. 2(c) shows the corresponding unwrapped surface by PUMF with  $p = 2$ . We can see that in a few iterations (eight) PUMF successfully accomplishes the unwrapping.

<sup>1</sup> The authors acknowledge Vladimir Kolmogorov for the max-flow/min-cut C++ code made available on the web.

---

**Algorithm 1** (PUMF) Graph cuts based phase unwrapping algorithm.

---

**Initialization:**  $\mathbf{k} \equiv \mathbf{k}' \equiv \mathbf{0}$ , possible\_improvement  $\equiv 1$

- 1: **while** possible\_improvement **do**
- 2:   Compute  $E(0,0), E(1,1), E(0,1)$ , and  $E(1,0)$  {for every horizontal and vertical pixel pairs}.
- 3:   Construct elementary graphs and merge them to obtain the main graph.
- 4:   Compute the min-cut  $(S, T)$   $\{S$ - source set;  $T$ -sink set}.
- 5:   **for all** pixel  $(i, j)$  **do**
- 6:     **if** pixel  $(i, j) \in S$  **then**
- 7:        $\mathbf{k}'_{i,j} = \mathbf{k}_{i,j} + 1$
- 8:     **else**
- 9:        $\mathbf{k}'_{i,j} = \mathbf{k}_{i,j}$  {remains unchanged}
- 10:    **end if**
- 11:   **end for**
- 12:   **if**  $E(\mathbf{k}'|\psi) < E(\mathbf{k}|\psi)$  **then**
- 13:      $\mathbf{k} = \mathbf{k}'$
- 14:   **else**
- 15:     possible\_improvement = 0
- 16:   **end if**
- 17: **end while**

---

Fig. 2(b) shows a wrapped phase image analogous to 2(a), but now the original phase corresponds to a (simulated) InSAR acquisition for a (real) high-relief mountainous area inducing, therefore, many discontinuities and posing a very difficult PU problem. Fig. 2(d) shows the unwrapped surface by PUMF. It should be stressed that this is a very tough phase unwrapping problem, and thus a quality map was supplied as an input discontinuity map to the algorithm. This quality map labels each difference as 0 or 1 according, respectively, to whether there is, or there is not, a discontinuity. PUMF accomplished the unwrapping, taking 17 iterations, with  $p = 2$ . With this setting, the accuracy of the unwrapping is given by an error norm of 0.0936 (squared rads). The error norm obtained with the WLS (Weighted Least Squares) algorithm [2] is 0.3977. This difference is due to the fact that the WLS algorithm relaxes the discrete problem to the continuum, before minimizing and then going back to the discrete domain, whereas the PUMF solves exactly the integer minimization problem. The error norm obtained with the LPN0 algorithm, the most (or among the most) accurate unwrapping technique known to date [2], was 0.0986, which confirms the outperforming accuracy of PUMF in this PU problem<sup>2</sup>.

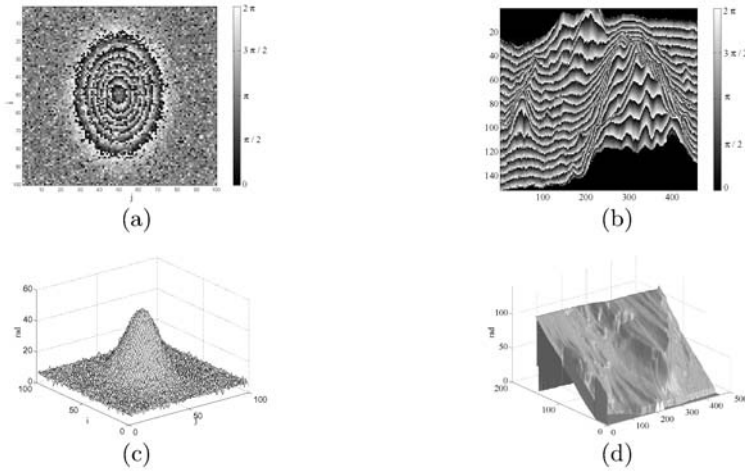
## 5 Concluding Remarks

We introduced a new PU algorithm that computes exactly the  $2\pi$  congruent minimum  $L^p$  norm of any linear function of the phase neighbouring differences,

---

<sup>2</sup> The error norms were calculated over the subset (of the entire image) defined by the quality map, plus a one pixel erosion in order to cut-off mask border pixels that usually have problems.





**Fig. 2.** (a) Wrapped phase image (rad) from a Gaussian absolute phase surface of height  $14\pi$  rad and standard deviations  $\sigma_i = 15$  and  $\sigma_j = 10$ . The correlation coefficient of the associated pair is  $\alpha = 0.8$ . (b) Wrapped phase image (rad) from a simulated InSAR acquisition for an area around Long’s Peak, Colorado (data distributed with book [2]). (c) Image in (a) unwrapped by PUMF (8 iterations). (d) Image in (b) unwrapped by PUMF (17 iterations).

for  $p \geq 1$ . This class of energy functions includes the usual norms used in PU and smoothing regularization functions used, for example, in image restoration.

The proposed algorithm is iterative, solving, in each iteration, a minimization over a binary *move* (each pixel allowed to remain unchanged or to be incremented by  $2\pi$ ). This minimization is implemented efficiently by exploiting recent results from [4] on energy minimization via max-flow/min-cut computations on certain graphs. In two experiments, the proposed PUMF algorithm outperformed state-of-the-art methods.

An open issue in the performance of PUMF for  $p < 1$ . In fact, we have noticed that, for this values of  $p$ , the algorithm is able to blindly, i.e., without supplying discontinuities explicitly, unwrap difficult examples. This is, however, an issue for future research.

## References

1. D. Ghiglia and L. Romero. Minimum  $L^p$  norm two-dimensional phase unwrapping. *Journal of the Optical Society of America*, 13(10):1999–2013, 1996.
2. D. Ghiglia and M. Pritt. *Two-Dimensional Phase Unwrapping. Theory, Algorithms, and Software*. John Wiley & Sons, New York, 1998.
3. J. Dias and J. Leitao. The  $\mathbb{Z}\pi\mathbb{M}$  algorithm for interferometric image reconstruction in SAR/SAS. *IEEE Transactions on Image Processing*, 11:408–422, April 2002.
4. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004.

5. P. Rosen, S. Hensley, I. Joughin, F. LI, S. Madsen, E. Rodriguez, and R. Goldstein. Synthetic aperture radar interferometry. *Proceedings of the IEEE*, 88(3):333–382, March 2000.
6. P. Jezard and R. Balaban. Correction for geometric distortion in echo-planar images from  $B_0$  field variations. *Magnetic Resonance in Medicine*, 34:65–73, 1995.
7. S. Pandit, N. Jordache, and G. Joshi. Data-dependent systems methodology for noise-insensitive phase unwrapping in laser interferometric surface characterization. *Journal of the Optical Society of America*, 11(10):2584–2592, 1994.
8. K. Itoh. Analysis of the phase unwrapping problem. *Applied Optics*, 21(14), 1982.
9. R. Goldstein, H. Zebker, and C. Werner. Satellite radar interferometry: Two-dimensional phase unwrapping. In *Symposium on the Ionospheric Effects on Communication and Related Systems*, volume 23, pages 713–720. Radio Science, 1988.
10. G. Nico, G. Palubinskas, and M. Datcu. Bayesian approach to phase unwrapping: theoretical study. *IEEE Transactions on Signal Processing*, 48(9):2545–2556, Sept. 2000.
11. B. Friedlander and J. Francos. Model based phase unwrapping of 2-d signals. *IEEE Transactions on Signal Processing*, 44(12):2999–3007, 1996.
12. D. Fried. Least-squares fitting a wave-front distortion estimate to an array of phase-difference measurements. *Journal of the Optical Society of America*, 67(3):370–375, 1977.
13. T. Flynn. Two-dimensional phase unwrapping with minimum weighted discontinuity. *Journal of the Optical Society of America A*, 14(10):2692–2701, 1997.
14. J. Leitão. Absolute phase image reconstruction: A stochastic non-linear filtering approach. *IEEE Transactions on Image Processing*, 7(6):868–882, June 1997.
15. J. Dias and J. Leitão. Simultaneous phase unwrapping and speckle smoothing in SAR images: A stochastic nonlinear filtering approach. In *EUSAR'98 European Conference on Synthetic Aperture Radar*, pages 373–377, Friedrichshafen, May 1998.
16. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, February 2004.
17. C. Jakowatz, D. Wahl, P. Eichel, D. Ghiglia, and P. Thompson. *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*. Kluwer Academic Publishers, Boston, 1996.

# A New Fuzzy Multi-channel Filter for the Reduction of Impulse Noise

Stefan Schulte, Valérie De Witte, Mike Nachtegael,  
Dietrich Van der Weken, and Etienne E. Kerre

Department of Applied Mathematics and Computer Science  
Ghent University, Krijgslaan 281 (Building S9)  
9000 Gent, Belgium

{Stefan.Schulte,Valerie.DeWitte,Mike.Nachtegael,  
Dietrich.VanderWeken,Etienne.Kerre}@UGent.be  
<http://fuzzy.ugent.be>

**Abstract.** One of the most common image processing tasks involves the removal of impulse noise from digital images. In this paper, we propose a new two step multi-channel filter. This new non-linear filter technique contains two separate steps: an impulse noise detection step and a noise reduction step. The fuzzy detection method is mainly based on the calculation of fuzzy gradient values and on fuzzy reasoning. This phase will determine three separate membership functions that will be used by the filtering step. Experiments prove that the proposed filter may be used for efficient removal of impulse noise from colour images without distorting the useful information in the image.

## 1 Introduction

A fundamental problem in image processing is to reduce effectively noise from a digital image while keeping its features intact. In this paper we mainly focus on filtering impulse noise from digital images. Impulse noise is usually characterized by some portion of image pixels that is corrupted, leaving the remaining pixels unchanged.

A digital colour image (denoted by  $O$ ) can be modelled in a certain colour space. As in most applications we use the RGB colour space. Colours in this model are represented by a three-dimensional vector, where each component is quantified to the range  $[0, 2^m - 1]$  (mostly with  $m = 8$ ). In practice a digital colour image  $O$  can be represented by a two-dimensional array of vectors where an address  $(i, j)$  defines a position in  $O$ , called a pixel or picture element. If  $O(i, j, 1)$  denotes the red component of a pixel at position  $(i, j)$  in an (noise-free) image  $O$  (respectively  $O(i, j, 2)$  the green and  $O(i, j, 3)$  the blue component), then we can model the occurrence of impulse noise for colour images as:

$$[A(i, j, 1) \ A(i, j, 2) \ A(i, j, 3)] = \begin{cases} [O(i, j, 1) \ O(i, j, 2) \ O(i, j, 3)] & \text{with prob. } 1 - pr \\ \text{noise pixel} & \text{with prob. } pr \end{cases}$$

where  $pr$  is the probability that a pixel is corrupted and where  $A$  is the corrupted image. An impulse noise pixel for colour images can be determined as follows:

$$\text{noise pixel} = \begin{cases} [p_k & O(i, j, 2) & O(i, j, 3)] \text{ or} \\ [O(i, j, 1) & p'_k & O(i, j, 3)] \text{ or} \\ [O(i, j, 1) & O(i, j, 2) & p''_k] \text{ or} \\ [O(i, j, 1) & p'_k & p''_k] \text{ or} \\ [p_k & p'_k & O(i, j, 3)] \text{ or} \\ [p_k & O(i, j, 2) & p''_k] \text{ or} \\ [p_k & p'_k & p''_k] \end{cases} \quad (1) \quad \text{with } \forall k \in \{1, \dots, n\}; \quad n \leq 2^m - 1$$

where  $p_k, p'_k$  and  $p''_k$  are integer values between 0 and  $2^m - 1$  (where  $m$  indicates the amount of bits used, possibly different, to store a colour pigment).

## 2 Fuzzy Impulse Noise Detection

In this paper we introduce a new two step filter called “Fuzzy Impulse noise Detection and Reduction Method for Colour images” (*FIDRMC*). In this section we explain the detection phase for the red component (the other two are similar) and in the next section the new filtering phase. This step uses (fuzzy) gradient values and a fuzzy rule (see GOA filter [1]) to determine if a certain pixel pigment is corrupted with impulse noise or not. This detection phase is an improved version of the one we have explained in [2].

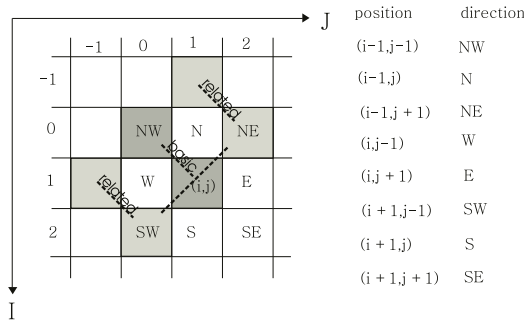
If  $A(i, j, 1)$  denotes the red component input image pixel at position  $(i, j)$ , then the definition of the gradient  $\nabla_{(k,l)}A(i, j, 1)$  becomes:

$$\nabla_{(k,l)} A(i, j, 1) = A(i + k, j + l, 1) - A(i, j, 1) \quad \text{with } k, l \in \{-1, 0, 1\},$$

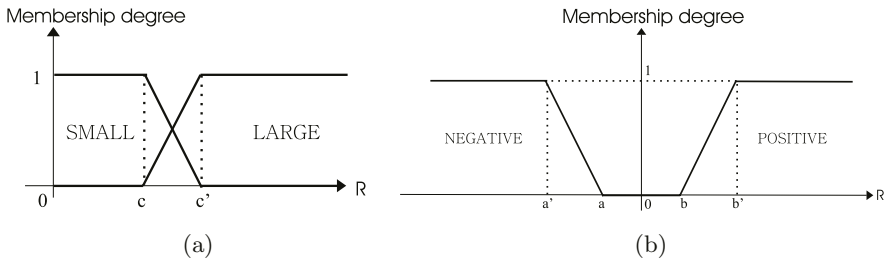
where the pair  $(k, l)$  corresponds to one of the eight directions  $R \in \{N = \text{North}, E = \text{East}, S = \text{South}, W = \text{West}, NW = \text{North West}, NE = \text{North East}, SE = \text{South East}, SW = \text{South West}\}$  w.r.t. the centre of the gradient  $(i, j)$  (see Fig. 1). For each direction  $R$  we calculate the corresponding gradient value that we simply call the basic gradient value ( $\nabla_R A(i, j, 1)$ ) and two related gradient values  $\nabla_R A(i', j', 1)$  and  $\nabla_R A(i'', j'', 1)$ . The centres of these two related gradient values  $((i', j')$  and  $(i'', j'')$ ) are making a right angle with the investigated direction  $R$ . Fig. 1 illustrates this principle for the North West direction (*NW*). The following fuzzy rule calculates the fuzzy gradient value (denoted by  $\nabla_R^F A(i, j, 1)$ ) for a direction  $R$  and centre  $(i, j)$ :

```

IF   |∇R A(i, j, 1)| is large AND |∇R A(i', j', 1)| is small
OR
     |∇R A(i, j, 1)| is large AND |∇R A(i'', j'', 1)| is small
OR
     ∇R A(i, j, 1) is positive AND (∇R A(i', j', 1) AND ∇R A(i'', j'', 1))
     are negative
OR
     ∇R A(i, j, 1) is negative AND (∇R A(i', j', 1) AND ∇R A(i'', j'', 1))
     are positive
THEN ∇RF A(i, j, 1) is large
    
```



**Fig. 1.** Involved centres for the calculation of the related gradient values in the *SE*-direction.



**Fig. 2.** The membership functions *SMALL* respectively *LARGE* (a), *NEGATIVE* respectively *POSITIVE* (b).

The basic gradient value will be high if impulse noise is present. Therefore it can be used to define impulse noise. But edges or contour pixels also have natural high gradient values and therefore we use the concept of fuzzy gradient values.

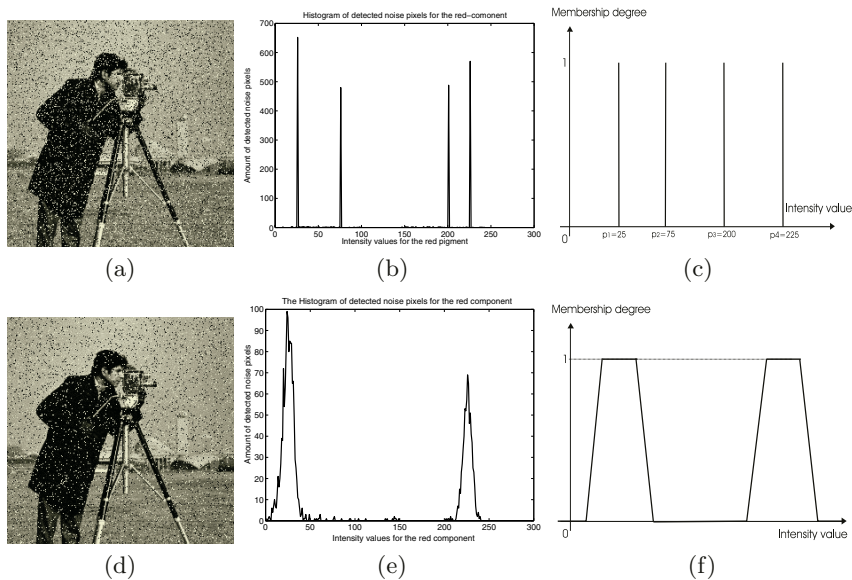
In fuzzy logic, features like “large”, “small”, “negative” and “positive” can be represented as fuzzy sets [3]. Fuzzy sets in turn can be represented by a membership function. Examples of the membership functions *LARGE* (for the fuzzy set *large*), *SMALL* (for the fuzzy set *small*), *POSITIVE* (for the fuzzy set *positive*) and *NEGATIVE* (for the fuzzy set *negative*) are shown in Fig. 2. The optimal parameters are [2]:  $c \in [50, 80]$ ,  $c' \in [110, 130]$ ,  $a' \in [-30, -20]$ ,  $a \in [-20, -15]$ ,  $b \in [15, 20]$ ,  $b' \in [20, 30]$ .

Finally we use the following fuzzy detection rule to decide if a certain pixel is (impulse) noisy or not:

**IF** most of the eight quantities  $\nabla_R^F A(i, j, 1)$  are *large*

**THEN** the central pixel pigment  $A(i, j, 1)$  is an impulse noise pixel pigment

This rule will be translated as: if for a certain central pixel  $(i, j)$  more than half of the fuzzy gradient values (thus more than four for a non-border pixel) are part of the (weak)  $\alpha$ -level of the fuzzy set *large*, then we can conclude that this pixel is an impulse noise pixel. The (weak)  $\alpha$ -level of a certain fuzzy set



**Fig. 3.** (a) A cameraman image corrupted with 20% impulse noise  $((p_1, p_2, p_3, p_4) = (25, 75, 200, 225))$  (b) the corresponding histogram of the detected noisy pixels and (c) the corresponding membership function *impulse noise*. (d) A cameraman image corrupted with 10% impulse noise  $((p_1, p_2) = (25, 225))$  plus Gaussian noise with  $\sigma = 5$ . (e) The corresponding histogram of the detected impulse noisy pixels and (f) the corresponding membership function *impulse noise*.

$FS$  [3] is the crisp set of all points in the universe of discourse  $U$  such that the membership function of  $FS$  is greater than or equal to  $\alpha$ . In order to be as critical as possible we keep this parameter very low [2]:  $\alpha \in [0, 0.1]$ . If we have decided that a certain pixel  $(i, j)$  is corrupted with impulse noise then we store the corresponding intensity value in a histogram. This histogram is used to define the membership function *impulse noise* (denoted respectively by  $\mu_{impulse}^{red}$ ,  $\mu_{impulse}^{green}$  and  $\mu_{impulse}^{blue}$  for the red, green and blue component). Two examples are presented in Fig. 3. For more details we refer to [2].

### 3 Filtering Phase

In contrast to other well known colour filters our new filtering step is not based on intensity values but on the differences between intensity values in the different components. The differences are used for filtering. This is realised by using the following matrices:

$$\begin{aligned}
 RG(i, j) &= A(i, j, 1) - A(i, j, 2) & RB(i, j) &= A(i, j, 1) - A(i, j, 3) \\
 GR(i, j) &= -RG(i, j) & GB(i, j) &= A(i, j, 2) - A(i, j, 3) \\
 BR(i, j) &= -RB(i, j) & BG(i, j) &= -GB(i, j)
 \end{aligned}$$

The first iteration is illustrated in Fig. 4 for the red component only (the green and blue components are filtered in the same way). If the output image  $F$  is the same as the input image  $A$  then the filter method is called recursively otherwise we call it non-recursively. In this pseudo-code (Fig. 4) we also use membership functions *impulse noise* for the defined matrices. These membership functions are defined by conjunction. For example, the membership degree  $\mu_{RG}$  is the conjunction of the membership degrees  $\mu_{impulse}^{red}$  and  $\mu_{impulse}^{green}$ . In fuzzy logic triangular norms and co-norms are used to represent conjunctions [3] (roughly the equivalent of AND operators) and disjunctions (roughly the equivalent of OR operators).

## 4 Experimental Results

Finally we present some experimental results. We compare our method with other well known fuzzy filters: FIRE [4] (fuzzy inference rule by else-action), DSFIRE [5] (dual step FIRE), PWLFIRE [6] (piecewise linear FIRE), AWFM [7] (adaptive weighted fuzzy mean), HAF [8] (histogram adaptive fuzzy), FMF [9] (fuzzy median filter), IFCF [10] (iterative fuzzy control based filter), FSB [11] (fuzzy similarity filter), FIDRM [2] (fuzzy impulse noise detection and reduction method). We also compare all these fuzzy filters with the following non-fuzzy filters: CWM [12] (centre weighted median), TSM [13] (tri-state median filter) and the LUM [14] (lower-upper-middle filter).

As a measure of objective dissimilarity between a filtered image and the original one we use the peak signal to noise ratio (PSNR Eq. 2 (in decibels dB)):

$$PSNR(Img, O) = 10 \log_{10} \frac{3NMS^2}{\sum_{c=1}^3 \sum_{i=1}^N \sum_{j=1}^M [O(i, j, c) - Img(i, j, c)]^2} \quad (2)$$

where  $O$  is the original image,  $Img$  the filtered image of size  $NM$  and  $S$  the maximum possible pixel value (with 8-bit integer values the maximum will be 255). Although this measure has his shortcomings w.r.t. expressing the quality of an image as observed by human beings, they are still widely used in the image processing community [15]. In order to get a clear idea of the performance w.r.t. the level of impulse noise, experiments have been carried out for 5%, 10%, 15% and 20% of impulse noise. This is illustrated in Table 1 where the numerical results for the test image *Trees* of size  $258 \times 350$  are shown. The test image is included in the Matlab package. Fig. 5 finally illustrates the main improvement for a part of a coloured image. The main improvement can be observed in regions with many details. Other filters introduce red artefacts at the leaves while this new filter performs very well. Other advantages of the FIDRM filter are: it doesn't blur (in contrast to HAF and IFCF) and it doesn't destroy useful information (see the line in Fig. 5).

**Input:**  $A = (A(i, j, 1), A(i, j, 2), A(i, j, 3))$ : the noisy colour image with impulse noise.

$\mu_{impulse}^{red}(A(i, j, 1))$ : the membership degree for the fuzzy set *impulse noise* for the red component image at position  $(i, j)$ .

$\mu_{impulse}^{green}(A(i, j, 2))$ : the membership degree for the fuzzy set *impulse noise* for the green component image at position  $(i, j)$ .

$\mu_{impulse}^{blue}(A(i, j, 3))$ : the membership degree for the fuzzy set *impulse noise* for the blue component image at position  $(i, j)$ .

$F(i, j, 1)$ : the red component output pixel at position  $(i, j)$ .

```

(1) FOR each non-border pixel  $(i, j)$ 
(2) IF  $A(i, j, 1) \in$  (weak)  $\alpha$ -level(fuzzy set impulse noise)
(3)    $s1_{RG} = 0, s2_{RG} = 0, s1_{RB} = 0, s2_{RB} = 0, s1_{RED} = 0$  and  $s2_{RED} = 0$ 
(4)   FOR  $h$  from  $-K$  to  $+K$ 
(5)     FOR  $l$  from  $-L$  to  $+L$ 
(6)        $s1_{RG} = s1_{RG} + (1 - \mu_{RG}(RG(i + h, j + l))) * RG(i + h, j + l)$ 
(7)        $s2_{RG} = s2_{RG} + 1 - \mu_{RG}(RG(i + h, j + l))$ 
(8)        $s1_{RB} = s1_{RB} + (1 - \mu_{RB}(RB(i + h, j + l))) * RB(i + h, j + l)$ 
(9)        $s2_{RB} = s2_{RB} + 1 - \mu_{RB}(RB(i + h, j + l))$ 
(10)       $s1_{RED} = s1_{RED} + (1 - \mu_{impulse}^{red}(A(i + h, j + l, 1))) * A(i + h, j + l, 1)$ 
(11)       $s2_{RED} = s2_{RED} + 1 - \mu_{impulse}^{red}(A(i + h, j + l, 1))$ 
(12)    END
(13)  END
(14)   $cor_{RG} = \frac{s1_{RG}}{s2_{RG}}$  and  $cor_{RB} = \frac{s1_{RB}}{s2_{RB}}$  and  $res_{RED} = \frac{s1_{RED}}{s2_{RED}}$ 
(15)   $res_{RG} = \mu_{impulse}^{green}(A(i, j, 2)) (A(i, j, 2) + cor_{RG})$ 
(16)   $res_{RB} = \mu_{impulse}^{blue}(A(i, j, 3)) (A(i, j, 3) + cor_{RB})$ 
(17)   $help = \frac{res_{RG} + res_{RB}}{\mu_{impulse}^{green}(A(i, j, 2)) + \mu_{impulse}^{blue}(A(i, j, 3))}$ 
(18)   $help = \max(\min(help, 2^m - 1), 0)$ 
(19)   $mem = \min(\mu_{impulse}^{green}(A(i, j, 2)), \mu_{impulse}^{blue}(A(i, j, 3)))$ 
(20)   $F(i, j, 1) = mem * res_{RED} + (1 - mem) * help$ 
(21) ELSE
(22)    $F(i, j, 1) = A(i, j, 1)$ 
(23) END IF
(24) END FOR

```

**Fig. 4.** Pseudo-code of the first filtering iteration for the red component.

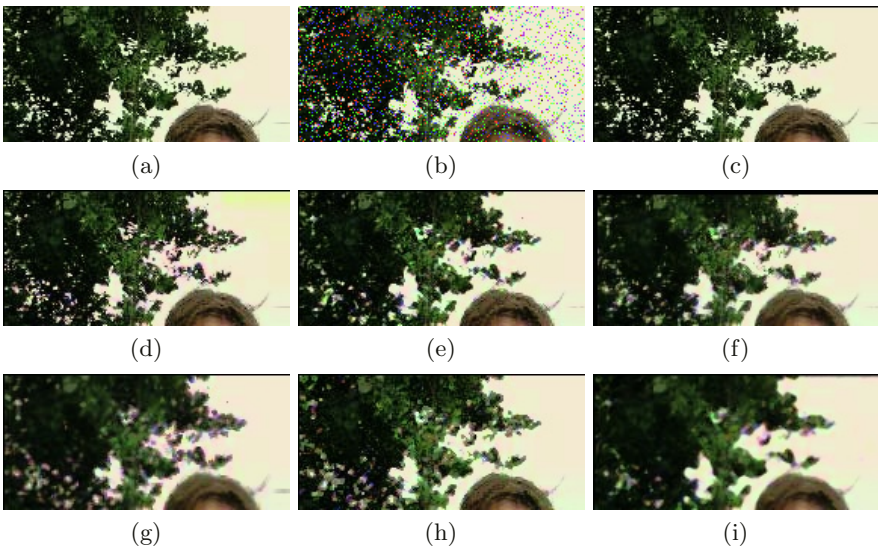
## 5 Conclusion

A new colour filter (FIDRMC), which is based on fuzzy logic has been presented. This filter is especially developed for reducing all kinds of impulse noise from digital colour images. A numerical measure, such as the PSNR (Eq. 2), and visual observations (Table 1 and Fig. 5) show convincing results for colour images.



**Table 1.** PSNR results for the (258 × 350-) *trees* image for different impulse noise levels (5%, 10%, 15%, 20%) and different filters.

	PSNR (dB)					PSNR (dB)			
	5%	10%	15%	20%		5%	10%	15%	20%
<i>Original</i>	17.02	14.04	12.27	11.08	<i>Original</i>	17.02	14.04	12.27	11.08
CWM (3 × 3)	27.76	26.17	25.59	25.13	TSM (3 × 3)	28.58	27.32	25.28	24.73
LUM	27.48	26.87	25.94	24.81	FSB	27.21	26.56	25.76	24.53
HAF	25.71	25.41	25.15	24.81	FIRE	30.42	26.93	23.90	21.35
AWFM	27.21	25.31	23.98	21.76	IFCF	27.46	26.65	25.62	24.56
DSFIRE	29.74	28.72	27.67	25.93	PWLFIRE	31.71	27.41	23.62	20.56
FMF	32.91	29.81	27.29	24.84	FIDRM	35.87	33.35	31.38	29.81
FIDRMC	42.99	39.98	37.87	36.02					



**Fig. 5.** The restoration of a part of a realistic colour image (a) corrupted with 10% impulse noise (b). The applied methods are: (c) FIDRMC, (d) component based FIDRM, (e) component based FMF, (f) component based DSFIRE, (g) component based HAF (h), component based AWFM, (i) component based IFCF.

## References

1. Van De Ville, D., Nachttegael, M., Van der Weken, D., Kerre, E.E., Philips, W.: Noise reduction by fuzzy image filtering. *IEEE T. Fuzzy Syst.* **11** (2001) 429-436
2. Schulte, S., Nachttegael, M., De Witte, V., Van der Weken, D., Kerre, E.E.: A new two step color filter for impulse noise. *Proceedings East West Fuzzy Colloquium* (2004) 185-192
3. Kerre, E.E.: *Fuzzy sets and approximate Reasoning*. Xian Jiaotong University Press, Softcover (1998).

4. Russo, F., Ramponi, G.: A Fuzzy Filter for Images Corrupted by Impulse Noise. *IEEE Signal Proceedings Letters* **3** (1996) 168-170.
5. Russo, F., Ramponi, G.: Removal of impulse noise using a FIRE filter. *Third IEEE Intern. Conf. on Image Processing* (1996) 975-978.
6. Russo, F.: Fire Operators for Image Processing. *Fuzzy Set. Syst.* **103** (1999) 265-275
7. Lee, C.S., Kuo, Y.H.: Adaptive fuzzy filter and its application to image enhancement. IN: Kerre, E.E., Nachttegael, M. (eds.): *Fuzzy Techniques in Image Processing*, Vol. 52, Springer Physica Verlag, Berlin Heidelberg New York (2000) 172-193
8. Wang, J.H., Chiu, H.C.: An adaptive fuzzy filter for restoring highly corrupted images by histogram estimation. *Proceedings of the National Science Council - Part A* **23** (1999) 630-643
9. Arakawa, K.: Median filter based on fuzzy rules and its application to image restoration. *Fuzzy Set. Syst.* **77** (1996) 3-13
10. Farbiz, F., Menhaj, M.B.: A fuzzy logic control based approach for image filtering. IN: Kerre, E.E., Nachttegael, M. (eds.): *Fuzzy Techniques in Image Processing*, Vol. 52, Springer Physica Verlag, Berlin Heidelberg New York (2000) 194-221
11. Kalaykov, I., Tolt, G.: Real-time image noise cancellation based on fuzzy similarity. IN: Nachttegael, M., Van der Weken, D., Van De Ville, D., Kerre, E.E. (eds.): *Fuzzy Filters for Image Processing*, Vol. 122 Springer Physica Verlag, Berlin Heidelberg New York (2003) 54-71
12. Ko, S.J., Lee, Y.H.: Center weighted median filters and their applications to image enhancement. *IEEE T. Circ. Syst.* **38** (1991) 984-993
13. Chen, T., Ma, K.K., Chen, L.H.: Tri-state median filter for image denoising. *IEEE T. Image Process.* **8** (1999) 1834-1838
14. Hardie, R.C., Boncelet, C.G.: LUM filters: a class of rank-order-based filters for smoothing and sharpening. *IEEE T. Signal Proces.* **41** (1993) 1834-1838
15. Van der Weken, D., Nachttegael, M., Kerre, E.E.: Using similarity measures for histogram comparison. *Lecture Notes in Computer Science.* **2715** (2003) 396-403

# Enhancement and Cleaning of Handwritten Data by Using Neural Networks

José Luis Hidalgo<sup>1</sup>, Salvador España<sup>1</sup>, María José Castro<sup>1</sup>, and José Alberto Pérez<sup>2</sup>

<sup>1</sup> Departamento de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia, Valencia, Spain  
{jhidalgo, mcastro, sespana}@dsic.upv.es

<sup>2</sup> Departamento de Informática de Sistemas y Computadores  
Universidad Politécnica de Valencia, Valencia, Spain  
aperez@disca.upv.es

**Abstract.** In this work, artificial neural networks are used to clean and enhance scanned images for a handwritten recognition task. Multilayer perceptrons are trained in a supervised way using a set of simulated noisy images together with the corresponding clean images for the desired output. The neural network acquires the function of a desired enhancing method. The performance of this method has been evaluated for both noisy artificial and natural images. Objective and subjective methods of evaluation have shown a superior performance of the proposed method over other conventional enhancing and cleaning filters.

**Keywords:** handwritten recognition, form processing, image enhancement, image denoising, artificial neural networks

## 1 Introduction

The field of offline handwritten recognition has been a topic of intensive research for many years [1–4]. One of the first steps in the classical architecture of a handwritten text recognizer is preprocessing, where noise reduction and normalization takes place. Preparing clean and clear images for the recognition engines is often taken for granted as a trivial task that requires little attention. However, this step undoubtedly influences the overall performance of the system. Neural networks for cleaning and enhancing scanned handwritten images are proposed in this work. For a review of image processing with neural networks, see [5].

There exist several methods to design forms with fields to be filled in. For instance, fields may be surrounded by bounding boxes, by light rectangles or by guiding rulers. These methods specify where to write and, therefore, minimize the effect of skew and overlapping with other parts of the form. These guides can be located on a separate sheet of paper that is located below the form or they can be printed directly on the form. The use of guides on a separate sheet is much better from the point of view of the quality of the scanned image, but requires giving more instructions and, more importantly, restricts its use to tasks where this type of acquisition is used. Guiding rulers printed on the form are more commonly used for this reason. Light rectangles can be removed more easily with filters than dark lines whenever the handwritten text touches the rulers. Nevertheless, other practical issues must be taken into account:

- The best way to print these light rectangles is in a different color (i.e. light yellow); however, this approach is more expensive than printing gray rectangles with black-and-white laser printers.
- A more economical and easier approach is to use gray rectangles printed by a black-and-white laser printer. This produces a pattern of pixels that is more difficult to remove.
- Very different types of handwriting instruments and different colors are used by different users.

The work described here consists of filtering the background noise caused mainly by gray rectangles used as guiding rulers. The proper elimination of these rectangles makes it possible to use this approach in the design of forms to be used by handwritten recognition systems, which is much cheaper than other approaches.

In many handwritten recognition systems, preprocessing does not require a binarization step. For this reason, the images should be maintained in gray-level quality. The enhancement of images should also correct traces with low, non uniform ink level produced by some handwriting instruments (such as some ball pens and pencils), which may be broken or disappear in the preprocessing.

## 2 The Spartacus Database

A new offline handwritten database for the Spanish language, which contains full Spanish sentences, has recently been developed: the Spartacus database [6] (which stands for *SPANish Restricted-domain TAsk of CURsive Script*). There were two main reasons for creating this corpus. First of all, most databases [7–12] do not contain Spanish sentences, even though Spanish is a widespread major language. Another important reason was to create a corpus from semantic-restricted tasks. These tasks are commonly used in practice and allow the use of linguistic knowledge beyond the lexicon level in the recognition process. The database includes 1 500 forms produced by the same number of writers, scanned at 300 dpi. A total of around 100 000 word instances out of a vocabulary of around 3 300 words occur in the collection.

As the Spartacus database consisted mainly of short sentences and did not contain long paragraphs, the writers were asked to copy a set of sentences in fixed places: dedicated one-line fields in the forms. Figure 1 shows one of the forms used in the acquisition process. These forms also contain a brief set of instructions given to the writer.

## 3 Cleaning and Enhancing Method

There are several classic spatial filters for reducing or eliminating high-frequency noise from images. The mean filter, the median filter and the closing/opening filter are frequently used [13]. The mean filter is a low-pass or smoothing filter that replaces the pixel values with the neighborhood mean. It reduces the image noise but blurs the image edges. The median filter calculates the median of the pixel neighborhood for each pixel, thereby reducing the blurring effect. Finally, the opening/closing filter is a mathematical morphological filter that combines the same number of erosion and dilation morphological operations in order to eliminate small objects from images [14, 15].

Squared neighborhoods of  $3 \times 3$  pixels with center at the modified pixel were employed in the filter implementations. However, the obtained images were not satisfactory enough (see Figure 2). For this reason, neural network filters [5] were used. Neural networks were used to estimate the gray level of one pixel at a time. The input to the network consists of a square of pixels centered at the pixel to be cleaned (see Figure 3).

ADQUISICIÓN DE ESCRITURA MANUSCRITA. Proyecto TIC-2006-1155 Código: 054-V  
 Esta muestra de escritura manuscrita servirá para ayudar a realizar y verificar sistemas de reconocimiento de escritura por ordenador. Por favor, escriba utilizando la zona sombreada como referencia, procurando no tocar la frase a copiar ni la línea inferior. Si le falta espacio, no hace falta que termine la frase.

¿Qué ríos nacen en Cantabria?

¿Qué ríos nacen en Cantabria?

Dime lo grande que es el Ebro.

Dime lo grande que es el Ebro.

Dime el río de menor longitud de Cataluña.

Dime el río de menor longitud de Cataluña.

Deseo saber el caudal del río Miño.

Deseo saber el caudal del río Miño.

¿Por cuántas comunidades pasa el Ebro?

¿Por cuántas comunidades pasa el Ebro?

¿Qué ríos hay en Asturias?

¿Qué ríos hay en Asturias?

Una habitación tranquila a nombre del señor Carpio.

Una habitación tranquila a nombre del señor Carpio.

Se puso amarillo de un acceso de ictericia.

Se puso amarillo de un acceso de ictericia.

Despiértelos mañana a las cuatro, por favor.

Despiértelos mañana a las cuatro, por favor.

Dígale que tengo un recado de parte de su marido.

Dígale que tengo un recado de parte de su marido.

*“This handwritten sample is intended to help the experimentation and testing of computer handwriting recognition. Please, write using the guiding rectangle as reference, trying not to touch the typographic text nor the bottom horizontal rule. If there is not enough space, the sentence should be left unfinished.”*

**Fig. 1.** An example of a filled acquisition form and the translation of the instructions given for filling out the form.

## 4 Simulated Noisy Image Dataset

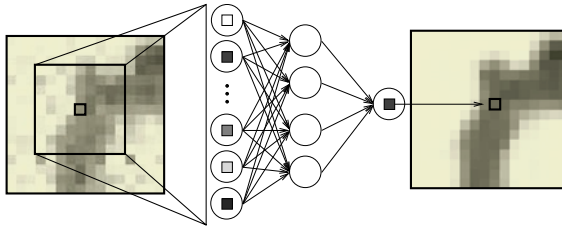
The main goal was to train a neural network in a supervised manner to obtain a clean image from a noisy one. In this particular case, it was much easier to obtain a simulated noisy image from a clean one than to clean a subset of noisy images.

The clean image database was obtained by scanning 150 white background handwritten sentences. The handwriting instrument was specially chosen in order to obtain uniform traces. The resolution was set to 300 dpi, which gives  $32 \cdot 10^6$  patterns. Pixels are codified as gray-levels in the interval  $[0,1]$ , where 0 means “black” and 1 means “white”.

The process for obtaining simulated noisy images follows the scheme presented in Figure 4. This process requires images of the background (gray rectangles) of the acquisition forms, which were obtained by printing and scanning the same background



**Fig. 2.** An example of an original scanned image (a) and the clean images obtained with the filters: Mean filter (b), Median filter (c), and Opening/Closing filter (d).



**Fig. 3.** Architecture of the artificial neural network to enhance images. The entire image is cleaned by scanning with the neural network.

of the original forms. First, in order to simulate the variability of the traces produced by some handwriting instruments (pencils, some ball pens, etc.), a trace noise was obtained by generating a white noise and applying an “oil” effect [16]. This trace noise was applied to the clean-trace image using the maximum operation, which only affects the ink and not the white background. Secondly, the noisy-trace image was combined with the scanned background noise to obtain the simulated noisy image. An example of a simulated noisy image is shown in Figure 5.

## 5 Enhancement and Cleaning with Neural Networks

### 5.1 Architecture

Multilayer perceptrons (MLPs) were used for the enhancement and cleaning of images. Only one output unit was needed to estimate the energy level (gray level) of the clean pixel. The activation function of the units of the hidden layer(s) was the sigmoid function, while the activation function of the output unit was the identity function. Due to the linear activation function, the output may be out of range, but, in practice, values were in the interval  $[0, 1]$ .

We employed the identity function at the output layer instead of the more commonly used sigmoid function because the characteristics of an MLP were improved significantly with the identity function when applied to regression problems such as image processing (see, for example, [17]). It should be noted that using a sigmoid activation function at the output layer is useful for applications where the output is in the form of binary values such as binarization image processing.

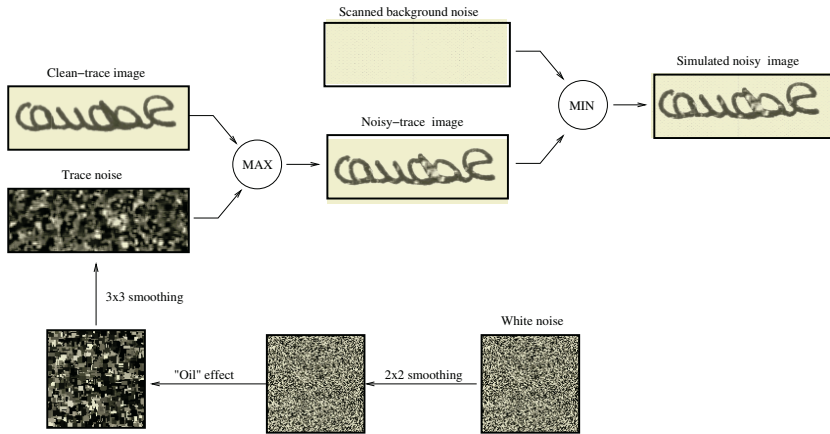


Fig. 4. Simulated noisy process.



Fig. 5. (a) Clean-trace image, and (b) Simulated noisy image.

The input units consisted of a squared window of pixels centered at the pixel to be cleaned. Neighborhoods from 2 to 5 were tested, where a neighborhood of  $n$  pixels means a squared  $(2n + 1)$ -sided input window to the MLP.

The entire image was cleaned by scanning all the pixels with the MLP. The MLP, therefore, functions like a nonlinear convolution kernel. The universal approximation property of a MLP guarantees the capability of the neural network to approximate any continuous mapping [18].

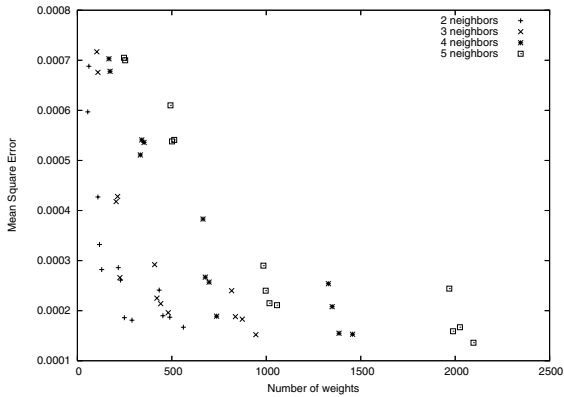
## 5.2 Training the Neural Networks

The obtained simulated noisy image corpus was divided into a training set, a validation set and a test set. The trained neural networks differed in the number of neighbor pixels (from 2 to 5), the number of hidden layers (one or two hidden layers) and the number of hidden neurons in each layer (from 2 to 16 hidden units). In every case, the online version of the backpropagation learning algorithm with momentum was used. For the same topology, several trainings were performed varying the learning rate, the momentum term and using different initializations of the weights. The stopping criteria was the mean squared error in the validation set.

## 6 Evaluation of the Cleaning and Enhancing Method

The proposed approach was objectively evaluated by using the simulated noisy image dataset. We measured the “closeness” of the original image (clean) and the cleaned

image (the simulated “noisy” image after being cleaned by each of the MLPs). This measure was obtained by calculating the mean squared error (MSE) between each pair of images in the test set. Figure 6 plots the MSE of all the trained MLPs. As can be observed, the best results were achieved with many different MLPs, demonstrating the robustness of the methodology. The best MLP (the one that obtained the lowest MSE in test set) used 5 neighbors at the input and two hidden layers of 16 and 8 units, respectively.



**Fig. 6.** Mean Squared Error of the test set for the trained MLPs. The number of neighbors and the complexity (number of weights) of the MLPs are displayed.

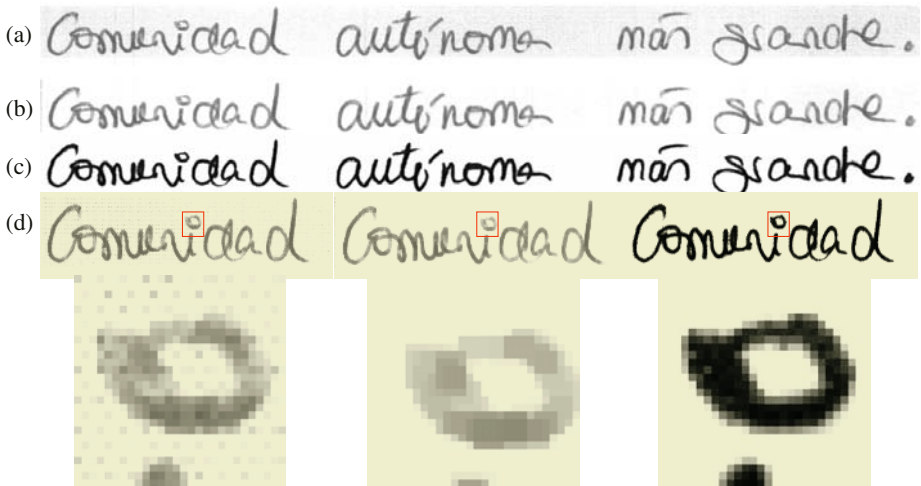
In order to perform a subjective evaluation of the cleaned Spartacus database, we visually inspected a subset of the cleaned images. An example of the performance of the proposed neural method, along with the result of the best used filter (opening/closing filter), is shown in Figure 7. As can be seen from the examples, the result clearly improves the image quality.

## 7 Summary and Conclusions

In this paper, we have described a generic cleaning and enhancing system for automatic form processing using neural networks. It takes clean and simulated noisy images to train and select the best neural network. Subjective and objective evaluations of the cleaning method show excellent results to clean forms with printed gray-areas to indicate where to fill in the information. The same idea could be used to clean and restore other types of images, such as noisy backgrounds in scanned documents, folded documents, stained paper of historical documents, vehicle license recognition, etc.

The proposed approach should also be evaluated objectively in a goal-directed manner [19], which means testing an image recognition system based on the results of our enhancing and cleaning method. We are planning to use both a standard HMM-based recognition system that has been developed in our research group and a commercial product. The purpose of using more than one recognizer in the evaluation is to prove that the improvement of performance brought about by the cleaning and enhancing procedure is independent of the features or methods that are used in the recognizers.





**Fig. 7.** (a) Original image, (b) result of applying the opening/closing filter, (c) result of applying the best MLP, and (d) Detail of the former images: Original image (left), result of applying the opening/closing filter (middle), and the result of applying the best MLP (right).

## Acknowledgments

Thanks to the Generalitat Valenciana under contract 20040479 for funding.

## References

1. Plamondon, R., Srihari, S.N.: On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. on PAMI* **22** (2000) 63–84
2. Bozinovic, R.M., Srihari, S.N.: Off-Line Cursive Script Word Recognition. *IEEE Trans. on PAMI* **11** (1989) 68–83
3. Bunke, H.: Recognition of Cursive Roman Handwriting – Past, Present and Future. In: 7th International Conference on Document Analysis and Recognition (ICDAR 2003), Edinburgh, Scotland (2003)
4. Toselli, A.H., et al.: Integrated Handwriting Recognition and Interpretation using Finite-State Models. *International Journal of Pattern Recognition and Artificial Intelligence* **18** (2004) 519–539
5. Egmont-Petersen, M., de Ridder, D., Handels, H.: Image processing with neural networks – a review. *Pattern Recognition* **35** (2002) 2279–2301
6. España, S., Castro, M.J., Hidalgo, J.L.: The SPARTACUS-Database: a Spanish Sentence Database for Offline Handwriting Recognition. In: IV International conference on Language Resources and Evaluation (LREC 04), Lisbon, Portugal (2004) 227–230
7. Guyon, I., Haralick, R.M., Hull, J.J., Philips, I.T.: Data Sets for OCR and Document Image Understanding Research. In Bunke, H., Wang, P.S.P., eds.: *Handbook of Character Recognition and Document Image Analysis*. World Scientific, Singapore (1997) 779–799
8. Hull, J.J.: A database for handwritten text recognition research. *IEEE Trans. on PAMI* **16** (1994) 550–554

9. Marti, U.V., Bunke, H.: The IAM-database: an English sentence adatabase for offline handwriting recognition. *International Journal on Document Analysis and Recognition* **5** (2002) 39–46
10. Suen, C.Y., et al.: Computer recognition of unconstrained handwritten numerals. *Special Issue of Proc IEEE* **7** (1992) 1162–1180
11. Viard-Gaudin, C., et al.: The IRESTE On/Off (IRONOFF) Dual Handwriting Database. In: *Fifth International Conference on Document Analysis and Recognition*, Bangalore, India (1999) 455–458
12. Wilkinson, R., et al.: The first census optical character recognition systems conference. In: *#NISTIR 4912, The U.S. Bureau of Census and the National Institute of Standards and Technology*, Gaithersburg, MD (1992)
13. Gonzalez, R., Woods, R.: *Digital Image Processing*. Addison-Wesley Publishing Company (1993)
14. Serra, J.: *Image Analysis and Mathematical Morphology*. Academic Press. New York (1982)
15. Serra, J.: *Image Analysis and Mathematical Morphology*. Vol: 2. Academic Press. New York (1988)
16. Holzmann, G.: *Beyond Photography*. Prentice Hall Professional Technical Reference (1988)
17. Suzuki, K., Horiba, I., Sugie, N.: Neural Edge Enhancer for Supervised Edge Enhancement from Noisy Images. *IEEE Trans. on PAMI* **25** (2003) 1582–1596
18. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press (1995)
19. Øivind Due Trier, Jain, A.K.: Goal-Directed Evaluation of Binarization Methods. *IEEE Trans. on PAMI* **17** (1995) 1191–1201

# Zerotree Wavelet Based Image Quilting for Fast Texture Synthesis

Dhammike S. Wickramanayake, Eran A. Edirisinghe, and Helmut E. Bez

Department of Computer Science, Loughborough University, UK  
D.S.Wickramanayake@lboro.ac.uk

**Abstract.** In this paper we propose a fast DWT based multi-resolution texture synthesis algorithm in which coefficient blocks of the spatio-frequency components of the input texture are efficiently stitched together (*quilted*) to form the corresponding components of the synthesised output texture. We propose the use of an automatically generated threshold to determine the significant coefficients which acts as elements of a matching template used in the texture quilting process. We show that the use of a limited set of, visually significant coefficients, regardless of their level of resolution, not only reduces the computational cost, but also results in more realistic texture synthesis. We use popular test textures to compare our results with that of the existing state-of-the-art techniques. Many application scenarios of the proposed algorithm are also discussed.

## 1 Introduction

Texture synthesis is particularly useful in modeling repetitive patterns such as human and animal skin, stone wood marble etc. A texture synthesis method starts from a sample image and attempts to produce a texture with a visual appearance similar to that sample, by repeated placement of micro patterns of texture elements on a surface so that when perceived by a human observer, it appears to be generated by the same underlying stochastic process. Unfortunately, creating a robust and general texture synthesis algorithm has been proven difficult.

The problem of synthesizing textures has been studied extensively and numerous approaches have already been proposed [1-5]. The inspiration for our work comes from the recent, efficient algorithms proposed by Efros & Freeman [6] and Lin Ling & C Liu [7]. Both these algorithms use patch based sampling and Lin Ling & C Liu [7] addresses the problem of constrained texture synthesis. These algorithms produce reasonably good quality results with less computation cost compared to the other algorithms. In Efros & Freeman's algorithm the output texture is formed by selectively transferring randomly selected blocks of a predefined size from the input texture image. Firstly, given that the top left hand corner block of the output image has been appropriately formed, a subset of blocks from which a good candidate for the block to its right (assuming a raster scanned order) could be found as follows: All possible blocks of the same block size from the input image is matched to the first block (top left hand corner) of the output image, under a certain overlap. Unfortunately this algorithm cannot be used for real time texture synthesis, as its efficiency is

relatively low. The use of exhaustive searching in choosing the best match causes computational power to be wasted. Due to the use of a random picking technique in selecting the final block to be patched with the preceding block, often the seam between the two adjacent blocks are quite visible. Even though a minimum error boundary cutting technique is used to smoothen off these sudden changes in texture, it involves computationally extensive methodologies such as dynamic programming and thus would not be suitable for real time applications.

In order to resolve the problems discussed above, in our previous work we proposed a Discrete Wavelet Transform (DWT) based multi resolution image quilting algorithm[8] in which coefficient-blocks of the spatio-frequency components of the input texture are efficiently *stitched* together to form the corresponding components of the synthesised output texture. In this paper we propose major improvements to this algorithm in terms of speed and the quality of synthesized texture. Important theoretical contributions made by Shapiro [9] to progressive encoding of images, namely, Embedded Zerotree Wavelet (EZW) coding, is modified and used in the proposed texture synthesis process.

For clarity of presentation the rest of the paper is divided into four further sections as follows. Section 2 discusses the possibility of using DWT in the analysis and the synthesis of a texture image and summarises our previous work in this area. Section 3 presents the proposed multiresolution framework. Section 4 provides experimental results and a comprehensive analysis of the results. Finally section 5 concludes, with an insight to possible improvements and future variations.

## 2 Wavelets in Analysis and Synthesis of a Texture Image

A texture image contains large amounts of perceptual data. Therefore the amount of bits required to represent one with good resolution is comparably high. Research in image compression technologies have proven that it is possible to produce a texture of near perceptual quality with only about 20-30 percent of total image data. Unfortunately, identifying this significant data in the pixel domain is difficult. However, images consist of a wide range of frequency components spread throughout the human visual frequency band. Some of these frequency components have a significant effect in human perception while some others have very low significance. Existing texture synthesis algorithms that produce *near photorealistic texture* demands high computational power often taking hours to synthesize small areas of texture. This is due to the reason that they are based on a texture analysis in the pixel domain. The best way to speed up these algorithms is to identify the perceptually significant frequency bands and use only those frequency components in the synthesis process. This requires a good frequency analysis method. In our previous work we successfully demonstrated the efficiency of using DWT as frequency analysis technique in texture synthesis [8]. However in this approach, we used only the lowest resolution sub-band, i.e.  $LL_3$  (assuming 3 levels of decomposition), together with one of  $LH_3$ ,  $HH_3$  or  $LH_3$ , in quilting two blocks in the synthesized texture. However as discussed above, there are perceptually important coefficients in sub-bands of higher resolution levels as well as perceptually negligible coefficients in low frequency bands. As this was not accounted for in the above texture synthesis process, the quality of the synthesized texture was

not very good for certain types of textures. Further the computational power was wasted in considering coefficients of insignificant visual impact in the synthesis process. In addition when using the L2 norm as the matching criteria, we do not get the visually best match, a problem inherited from Freeman's method. The method proposed in this paper overcomes all above shortcomings.

### 3 Proposed Multiresolution Framework

We commence with a three level DWT decomposition of the input sample texture and the output (to be synthesized) texture. The basic idea of proposed multiresolution texture synthesis algorithm is to synthesize each sub-band of the output texture by the corresponding sub-band of the input, sample texture. This texture synthesis procedure is described in detail below.

Let  $B_{s(x,y)}$  denote a general, block tree (see fig. 1(a), a combination of several adjacent wavelet coefficient trees with the root being a coefficient in LL3) of the decomposed sample (Note: subscript 's') image, located at position  $(x, y)$  relative to its origin. In our experiments we have set the block size to  $2^{5-l} \times 2^{5-l}$  (where  $l = 3$ , number of decomposition levels). We first pick a block tree,  $B_{s(x_1,y_1)}$  randomly and place it in the top left hand corner of the output coefficient image (see figure 1(a)). Let this block tree be denoted by  $B_{o(0,0)}$ . Subsequently we create a so-called matching mask tree from  $B_{o(0,0)}$  by only selecting the coefficients higher than a pre defined threshold out of the right hand side edge zone of the block  $B_{o(0,0)}$  (see figure 1(b)). Let this matching mask tree be denoted by  $M_{o(0,0)}$ . This matching mask tree (figure 1(e)) is then moved around the sample image's 3 level DWT representation in search of the best match. When the best match is found, the corresponding block tree (figure 1(d)) is picked and placed in the output representation (see figure 1(c)).

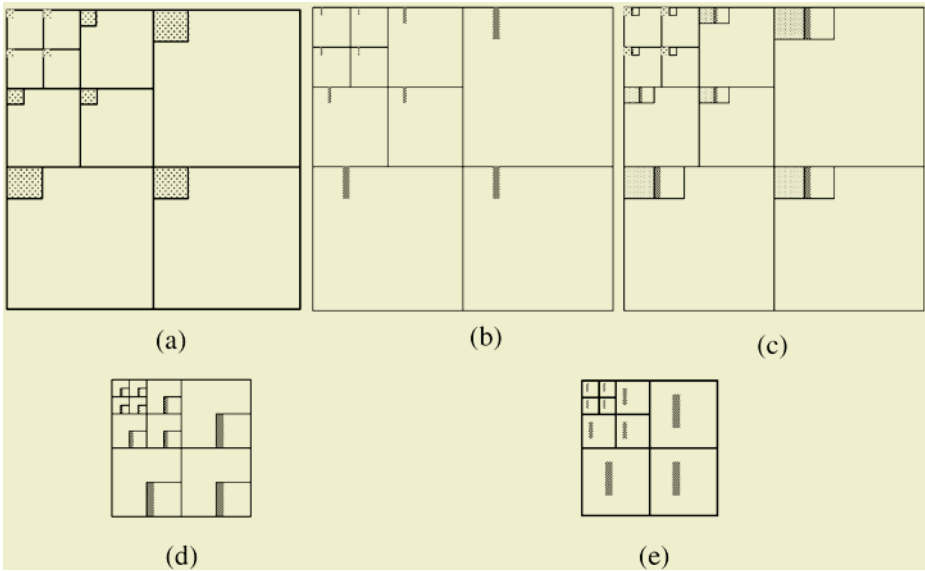
Note that due to the use of the matching mask tree, the coefficients above the pre defined threshold in all sub-bands are used in matching. All coefficients below the threshold were disregarded in matching, regardless of whether they came from a low resolution sub-band or not. The matching criterion used is described as follows.

In general, if  $B_{o(x_1,y_1)}$  and  $B_{s(x_2,y_2)}$  are two block trees to be matched, we say  $B_{s(x_2,y_2)}$  is the best match for  $B_{o(x_1,y_1)}$  if  $d(B_{o(x_1,y_1)}, B_{s(x_2,y_2)})$  is minimum for all possible  $B_s$  block trees where,

$$d(B_{s(x_2,y_2)}, B_{o(x_1,y_1)}) = \sum_{all\ i} [ \{ \partial B_{o(x_1,y_1)}(i) - \partial B_{s(x_2,y_2)}(i) \}^2 ] \quad (1)$$

Where  $\partial B_x$  is the *matching mask tree* of the *block tree*  $B_{s(x,y)}$  (see figure 1(c)) and  $i$  is the  $i^{th}$  element in the matching mask tree.

**Determining the Threshold:** Our algorithm is based on two important observations; (i) natural images in general have a low pass spectrum. When an image is wavelet transformed the energy in the sub-bands decrease as the scale decreases, so the wave-



**Fig. 1. Construction of the output texture.** (a) First random *block tree* (⋮) placed on top left hand corner of the output texture (b) *Edge zone tree* of the first *block tree*. (c) First random *block tree* (⋮) and its best match (◻) (second *block tree*) placed on top left hand corner of the output texture. (d) Best position is found and the corresponding *block tree* is picked (e) *Matching mask tree* moved around the sample texture to find the best match.

let coefficients will, on average, be smaller in the higher sub-bands than in the lower sub-bands. (i.e. higher sub-bands only add details). (ii) Larger wavelet coefficients are more important than the smaller coefficients. Therefore we need to find a threshold, which gives the minimum number of coefficients which could result in the best possible perceptual quality. Experiments carried out by us proved that the optimum threshold is dependant on the details contained in the texture. Thus we use the following method based on the largest coefficient in  $LL_3$  sub-band (this sub-band contains “the” most perceptually significant information). We use the following empirical formula to obtain the threshold,  $t$ ,

$$t = 2^{\lfloor \log_2(\text{MAX}(|LL_3(x,y)|)) \rfloor} / 10 \quad (2)$$

where  $\text{MAX}()$  means maximum coefficient value and  $LL_3(x, y)$  denotes a general coefficient in  $LL_3$  sub-band. Our experiments showed that when this threshold is used only 10%-15% of the coefficients are selected. These coefficients contain the perceptually significant frequency components.

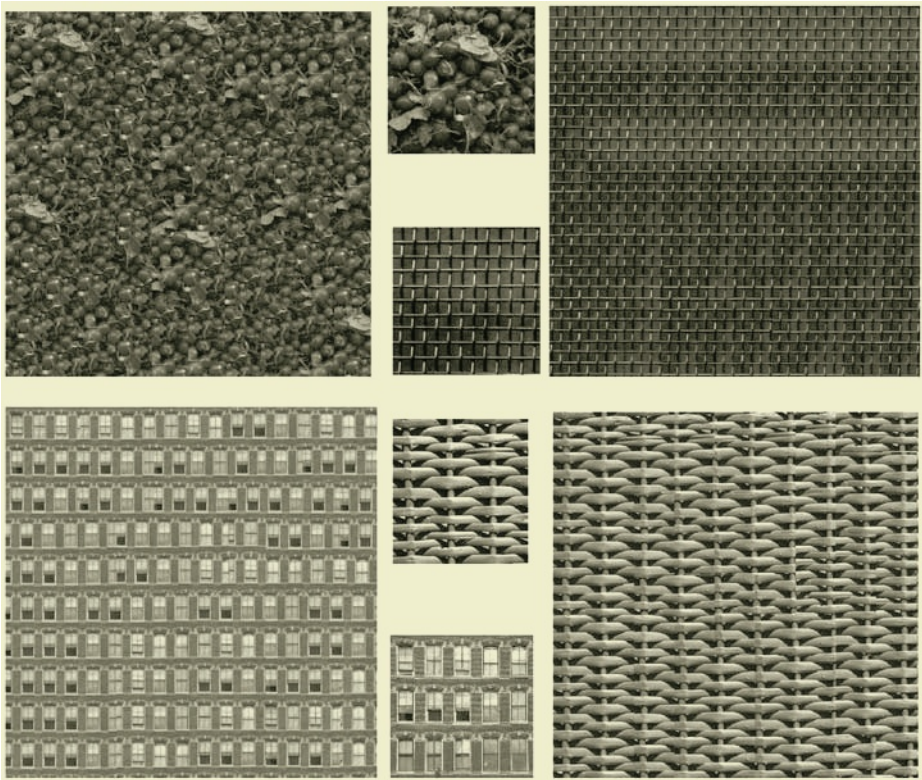
**Mask Creation and Best Match Selection:** Once the above threshold is used to filter out the significant coefficients, the matching mask tree is created to be the significant coefficient representation of the *edge zone tree* (see fig. 1(b)). This mask is overlapped with the sample coefficient image and moved vertically and horizontally to all

possible locations until the best match is found. At the best location, the block tree adjacent to the matching mask tree is picked and placed in the output representation. This process is repeated until whole output texture is filled.

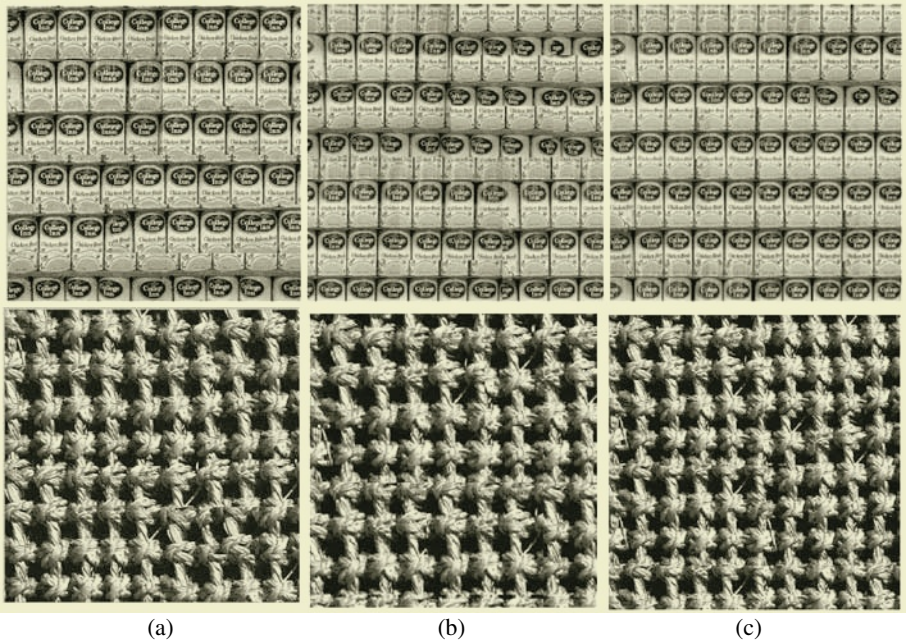
## 4 Experimental Results and Analysis

In order to analyse the performance of the proposed algorithm, experiments were performed on a widely used set of texture images, consisting of textures of both regular and stochastic nature. A typical set of sample images and the output textures obtainable using the proposed texture synthesis algorithm is illustrated in figure 2. The results clearly indicate that the proposed method is capable of providing high quality texture synthesis for a wide variety of textures. The selection of publicly available texture images for our experiments should enable readers to compare the performance of our algorithm with that of others.

Figure 3 compares the performance of proposed technique with that of Effros's [6] and our previous method based on DWT [8], for two regular test images. They clearly illustrate the improved subjective quality performance of the proposed algorithm.



**Fig. 2.** Synthesized (large) texture samples using the proposed algorithm. The corresponding small textures show the original texture samples.



**Fig. 3.** Subjective quality comparison between textures synthesized by (a) Effros's method (b) Our previous method (c) Proposed method.

#### 4.1 Detailed Analysis

In contrast to the method proposed by Effros and Freeman, we have adapted a multiresolution matching strategy in selecting the adjacent blocks of the output texture. The use of pixel level detail [6] would not only make the texture synthesis inefficient, but also unsuitable for real time texture synthesis capabilities expected from modern imaging applications. In our previous method, we adopted all coefficients from a fixed number of sub-bands from the lowest level of resolution, for comparisons. Those coefficients, in some instances excluded certain significant high frequency coefficients and included certain insignificant low frequency components. Yet it managed to achieve a remarkable speed in synthesis while maintaining reasonably good quality compared to Effros and Freeman algorithm. In the proposed algorithm when the threshold is lowered the numbers of coefficients included in the mask tree is increased. This in turn increases the quality of synthesized texture. In the experiments performed we started with the threshold at its maximum value,  $t \times 8$  and gradually decreased. Our observations were that the quality of the synthesized texture increased up to a certain maximum level and lowering the threshold further, does not significantly improve the quality of the final output texture. At the same time when the threshold is lowered the no of comparisons increase due to increased number of coefficients in the mask tree, resulting in increased computational cost. This increases the synthesis time. Consequently we need to find a trade off between quality and speed. Empirically we found  $t$  to be the best threshold.



In order to maintain the global structure of the overall texture it is important to select the block size as large as possible. This also accounts for increased efficiency of the algorithm as the choice of blocks available for filling the output texture becomes less, making the process fast. At the same time, selection of large block sizes makes it increasingly difficult to find overlapping areas providing a good match, lowering the quality of the resulting texture. Selection of the optimum size of the block is dependent on the repeating pattern contained in the texture to be synthesized. The use of small block sizes will increase the synthesis time. Thus in an effective implementation of the proposed algorithm we need to have a trade off between the image quality and efficiency in selecting the block size. Experiments have shown block size  $8 \times 8$  in  $LL_3$  sub-band gives better results for most of the textures.

In selecting the matching block, width of the matching mask tree (corresponds to the area of overlap of block trees to be matched) will also account for the quality and the speed of synthesis. Use of less number of overlapping elements (coefficients) results in increased efficiency and more visible artefacts at block boundaries. Increase of overlapping elements results in better quality with less artefacts and increased synthesis time. However, a too extensive increase in overlapping area will result in noticeable artefacts as it makes it more difficult for the algorithm to make the correct decision on the perceptually best matching block. In order to maintain a compromised situation we have adapted an overlap of a single coefficient row (or column) at level  $LL_3$ , of decomposition. This amounts to an overlap of 8 pixel rows (or columns) in the pixel domain.

## 4.2 Applications

The following is a summary of applications that could benefit from the multiresolution design of the proposed fast texture synthesis algorithm.

**Progressive 2D Texture Transmission:** Within a progressive transmission scenario, data is transmitted according to significance. The special design of the proposed texture synthesis algorithm allows DWT coefficient significance based progressive creation, transmission and reconstruction of the synthesized texture.

**Texture Mapping of Progressively Transmitted 3D Structures:** MPEG 4 AFX standard is currently working on progressive transmission of 3D structures. Initially they transmit data sufficient for a coarse representation of the structure. Our algorithm can complement this effort by texturing that surface with minimal transmission of texture data. Thus, both the structure as well as the texture can be refined progressively with more data transmission.

**Compressed Domain Texture Synthesis:** Synthesizing a compressed output texture with the use of a compressed original texture sample. Useful in fast, on-demand applications.

## 5 Conclusion

In this paper we have introduced a novel approach to synthesizing textures under a multi resolution framework. We have provided experimental results and an in-depth

analysis, proving that the proposed method works remarkably fast, producing better output texture quality as compared to the method proposed in [6,8]. The multiresolution nature of the proposed framework also makes it easily applicable to modern imaging applications needing progressive transmission capabilities.

In designing the above multi-resolution texture synthesis algorithm we have made a compromise between the synthesised texture quality and the algorithmic complexity by not performing seamless edge construction algorithms as in [6] and [7]. However due to the multi-resolution approach and the novel matching criteria adopted, we have managed to obtain perceptually equivalent (or better) synthesised texture quality to that of [6,7] at a much less computational complexity. We are currently looking at the implementation optimisation of the algorithms and the use of fast, simple, seamless edge cutting/construction algorithms. We are also in the process of applying the idea to handle the texture synthesis part omitted from consideration in the fast MESHGRID coding algorithm of [10], which has been a key contribution to the MPEG-4 AFX coding standard. This is expected to extend the applicability of the MESHGRID algorithm to full, fast, multi-scalable 3D object/surface coding.

## References

1. A.Witkin and M.Kass. Reaction-diffusion textures. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, July 1991.
2. A.Efros and T.Leung. "Texture synthesis by non-parametric sampling". In *International Conference for Computer Vision*, Volume 2, pages 1033-1038, September 1999.
3. Turk, G. "Generating textures on arbitrary surfaces using reaction-diffusion", In *Computer Graphics (SIGGRAPH '91 Proceedings)*, pages 289-298, July 1991.
4. Lewis, J.P. "Texture synthesis for digital painting" In *Computer Graphics (SIGGRAPH '84 Proc.)*, pages 245-252, July 1984.
5. Fournier,A., Fussel, D., and Carpenter, L. "Computer rendering of stochastic models" in *Commun, ACM*, pp. 371-384, June 1982.
6. A.Efros and W. T. Freeman, "Image Quilting for Texture Synthesis and Transfer", *Proceedings of SIGGRAPH '01*, pages 341-346, Los Angeles, California, August, 2001
7. L Ling, C E Liu, Ying Xing, Baining Guo, and Heung-Yeung Shum. "Real Time Texture synthesis by patch based Sampling" *ACM Transaction on Graphics*, Vol 20, No 3, July 2001, pp.127-150.
8. Wickramanayake, D.S., Edirisinghe, E.A. and Bez, H.E., "Fast Wavelet Transform Domain Texture Synthesis" , *Proceedings of the SPIE International Conference on Visual Communications & Image Processing* , 5308, VCIP 2004 , San Jose, California, January 2004, pp. 979-987,, ISBN: 0-8194-5211-4 .
9. Jerome M. Shapiro, "Embedded image coding using zero trees of wavelet coefficients", *IEEE Trans. On Signal processing*, Vol. 41, No. 12, December 1993, pp 3445-3462.
10. I.A.Salomie, A.Munteanu, A.Gavrilescu, G.Lafruit, P.Schelkens, R.Deklerck, J.Cornelis, 'MESHGRID – A Compact, Multiscalable and Animation-Friendly Surface Representation', *IEEE Trans. on CSVT*, Vol.14, No.7, July 2004, pp. 950-966.

# Semantic Feature Extraction Based on Video Abstraction and Temporal Modeling

Kisung Lee

Kongju National University, Kongju, Korea  
klee@kongju.ac.kr

**Abstract.** This paper presents a novel scheme of object-based video indexing and retrieval based on video abstraction and semantic event modeling. The proposed algorithm consists of three major steps; Video Object (VO) extraction, object-based video abstraction and statistical modeling of semantic features. Semantic feature modeling scheme is based on temporal variation of low-level features in object area between adjacent frames of video sequence. Each semantic feature is represented by a Hidden Markov Model (HMM) which characterizes the temporal nature of VO with various combinations of object features. The experimental results demonstrate the effective performance of the proposed approach.

## 1 Introduction

As the Internet grows explosively, needs of multimedia processing such as image, audio, and video data, are also increasing rapidly. Therefore analysis and representation of multimedia contents become more and more important issues. MPEG-7, the representation standard enables flexible access and manipulation functionalities of audio/visual data in a unified manner. To make it more useful and efficient for real-world applications, multimedia indexing and retrieval schemes have been emphasized as filtering and interfacing tools for this standard.

Although many works [1, 2] have been done in the area of content-based image indexing, video object(VO)-based area has not been fully explored. As the research of multimedia indexing advances more and more, demands for bridging the gap between low-level perceptual features and high-level concepts become increasingly important issues. By composing semantically meaningful features from various low-level features, it is possible to build more advanced indexing and retrieval system equipped with automatic annotation functionality. Currently, some recent works in video indexing area deal with both object-based approach and semantic feature composition [3-5].

Video abstraction is another important issue for high-level feature modeling. A video abstraction is defined to be a sequence of still or moving images (with or without audio) presenting the content of a video in such a way that the respective target group is rapidly provided with concise information about the content while the essence of the original message is preserved [6]. While it has been adapted to video skimming or browsing applications of most indexing and retrieval systems, we ap-

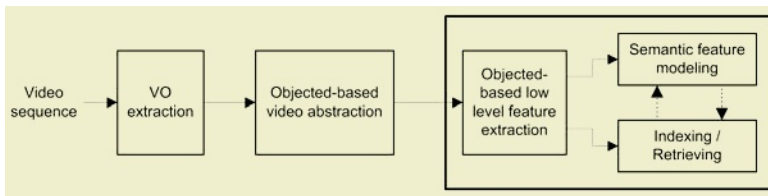
plied this technique as a filter for following feature extraction step, since it reduces temporal data redundancy of video sequence and helps semantic models to concentrate on apparent changes of object features in time domain as the object-based approach does similar function in spatial domain.

From the motivations described above, this paper describes an approach which integrates VO extraction, object-based abstraction, and high-level semantic feature modeling.

## 2 System Overview

Fig. 1 is the block diagram of the proposed approach.

First step is object segmentation to extract interesting area, i.e. VO, for indexing and retrieval. There are two different scenarios for the applications of content-based video extraction and analysis. One is a change-detection-based VO extraction algorithm that is appropriate for video sequences with stationary background. The other is an object tracking-based method, which is more suitable for video sequences with moving background. While the algorithm for stationary background is operated automatically, the latter is done by semi-automatic manner, i.e. the algorithm needs first frame with manually segmented object mask and starts tracking the object in the consecutive frames. In our approach, two novel algorithms proposed in [7] are exploited to cover both types of applications.



**Fig. 1.** Block diagram of object-based semantic feature indexing/retrieving system

After VO extraction, object-based video abstraction is performed, which can provide the abstracted manner to represent video sequence efficiently with all the gists in a video preserved. Main advantage of abstraction is to reduce the data redundancy from video shot and to provide reliable feature data for following feature extraction and modeling stage. The approach proposed in [8] was utilized for this step.

Final step is semantic feature extraction to bridge the gap between low-level feature and high-level concept to some extent. There are three components of blocks in this stage. At first, perceptual features such as shape, motion, etc are collected as input of semantic-level feature composition. Then, HMM is modeled and trained to create advanced features with these data. The dotted lines in Fig. 1 explain query process. In the case of query, video clips are commonly processed through extraction, abstraction and low-level feature extraction steps. Then, extracted low-level features are fed into previously trained probabilistic models and classified into nearest ones.

### 3 Semantic Feature Modeling

The main advantage of video features beyond image ones is that video features include the temporal information on the attributes of each video frame through the whole sequence. By characterizing temporal variations of features between objects in adjacent frames, semantic features can be modeled to some extent.

This section describes the proposed semantic feature composition method. Since we consider the algorithm for generalized applications, extracted VO needs to be processed for normalization prior to feature extraction. For perceptual features, we mainly focus on shape and motion of VO because they represent the characteristics of VO quite well if VO can be extracted precisely with acceptable noise. By composing these features with HMM, semantically meaningful event is composed, trained, and tested by query process. The following subsections present those procedures in detail.

#### 3.1 Preprocessing

In order to compensate the global camera motion dominating through whole frame, the area of extracted object is considered as “interesting domain” for features. Therefore we start from normalization of VO area in terms of object size and COG(Center of Gravity).

COG is calculated with Eq.(1) on the extracted object mask image  $B(x, y)$  which contains 0 for background and 1 for foreground.

$$x_{COG} = \frac{\sum_{x=1}^w \sum_{y=1}^h B(x, y) \cdot x}{\sum_{x=1}^w \sum_{y=1}^h B(x, y)} \quad y_{COG} = \frac{\sum_{x=1}^w \sum_{y=1}^h B(x, y) \cdot y}{\sum_{x=1}^w \sum_{y=1}^h B(x, y)} \quad (1)$$

where  $w$  and  $h$  are width and height of image  $B$  respectively.

Distance from COG to end points of  $x, y$  directions in each frame is calculated by Eq. (2) as follows.

$$\Delta_{x,t} = \max(x_{COG} - x_{\min,t}, x_{\max,t} - x_{COG}), \quad \Delta_{y,t} = \max(y_{COG} - y_{\min,t}, y_{\max,t} - y_{COG}) \quad (2)$$

where  $t=1, \dots, T$  and  $T$  is total frame size of abstracted video, and  $x_{\min,t}$  and  $y_{\min,t}$  are minimum  $x, y$  coordinate positions where object mask exists in frame  $t$ . And same expressions are applied to  $x_{\max,t}$  and  $y_{\max,t}$  as well.  $\Delta_{x,\max}$  and  $\Delta_{y,\max}$  which are maximum values of  $\Delta_{x,t}$  and  $\Delta_{y,t}$  for all  $t$ , are considered as normalized distance from COG in  $x, y$  direction. The original image frames are cropped in  $x, y$  direction with the length of  $\Delta_{x,t}$  and  $\Delta_{y,t}$  respectively, then are resampled into predefined VO area size so that all the video clips can be normalized with same size for further processing.

#### 3.2 Shape Features

##### 3.2.1 Moments of Boundary Segments

The shape of boundary segments is one of the major perceptual features that can represent the property of an object in an image. A semantic event can be modeled by

exploring the temporal variation of an object shape through the whole frames within a shot. The shape of boundary segments can be described quantitatively by using moments, entropy, and so on [9]. Let the amplitude of the shape boundary be a random variable  $r$ , and amplitude histogram  $p_r(x), x = 1, \dots, R$ . Then,  $i$ -th central moment of  $r$  is defined as

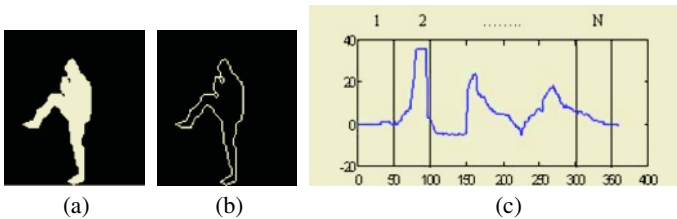
$$\mu_i = \sum_{x=1}^R (x - m)^i p_r(x), \quad i = 1, 2, \dots \tag{3}$$

where first moment of  $r$  is

$$m = \sum_{x=1}^R x p_r(x) \tag{4}$$

It has been well known that first few moments are required to differentiate between boundary signatures of clearly distinct shapes. The second moment  $\mu_2$  has been regarded as a measurement of the spread of the curve about the mean value of  $r$  and the third moment  $\mu_3$  measures its symmetric property with respect to its mean value.

As shown in Fig. 2, after extraction of boundary signature of an object, it is divided into  $N$  equally spaced 1-D blocks for each frame. Therefore the signature divided by each block can be treated as a boundary segment. The first moment (Eq. (4)), and second and third central moments (Eq. (3)) are calculated for each boundary segment. By considering moments in multiple segments, we can put spatially correlated features together and produce more reliable features than considering whole object boundary as one segment at the expense of number of features.



**Fig. 2.** Moments feature extraction from segments of boundary signature. (a) binary mask of normalized object, (b) extracted boundary, (c) segmentation of 1-D boundary signature: x axis is degree of angle, y axis is relative value of the boundary signature

**3.2.2 UFF(UNL Fourier Features)**

Drawback of moments of boundary signature is that they are applicable to only some part of the pattern if the object has holes inside the object area or deep bays along with its boundary. For instance, in Fig. 2, the concave boundary of lifted leg are an insurmountable limit for these features. Well-known Fourier descriptors(FD) also have similar limitations as signatures. A further drawback of FDs are that the patterns are limited to only closed curves.

In order to compensate these drawbacks, more generalized shape descriptors called UFF(UNL Fourier Features) are applied as the second shape features. A method denoted as UNL transform, performs a normalization operation for translation and scale changes and causes rotations to appear as periodic translations in the transformed

representations of the pattern. This procedure creates an optimal input for a 2-D Fourier image transform which yields the numerical descriptors called UNL Fourier Features, and consequently permits the mapping of any shape to a single vector (or point in an n-dimensional space). The definition and properties of UFF in detail is described in [10].

Besides its general-purpose applicability to various kinds of 2-D patterns including open, multiple curves, and curves with holes, UFF has been reported to outperform other boundary-based shape descriptors in image processing area by some experimental works [11].

### 3.3 Motion Features

The procedure for motion feature extraction consists of two steps. In first step, motion vectors are extracted for each pixel by hierarchical block matching (HBM) algorithm [12] as shown in Fig. 3 (c). Since we regard object mask area as only interesting domain, it is truncated by Eq.(5). (See Fig. 3(d))

$$\begin{aligned} MV_{obj}(x, y) &= MV(x, y) \text{ if } NR(x, y) > 0, \\ MV_{obj}(x, y) &= 0 \text{ otherwise,} \end{aligned} \quad (5)$$

where  $NR(x,y)$  is normalized object image produced by preprocessing step.

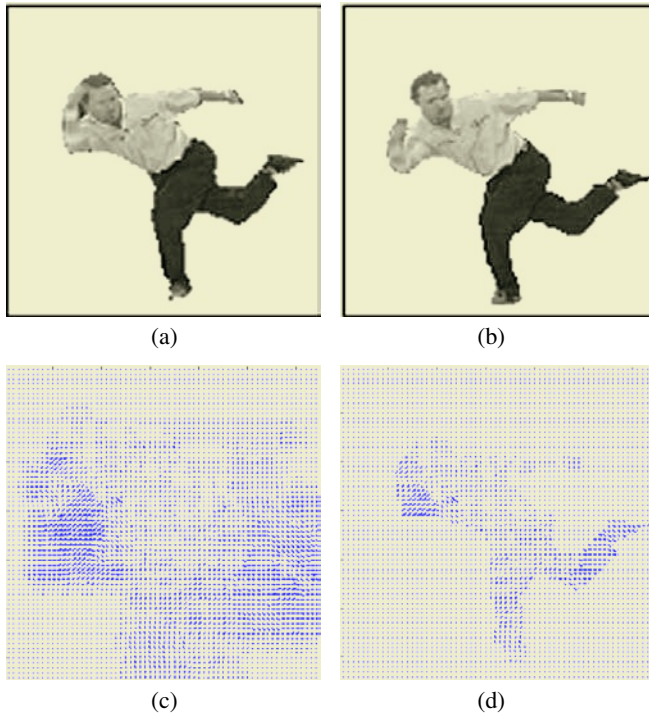
Now, the  $MV_{obj}$  is divided into equally spaced  $M \times N$  regions so that it can preserve spatially correlated motion features together. Motion vectors of x and y direction are considered as different features. Therefore total  $2MN$  features per frame are extracted for motion. First moments of x, y motion vectors in each segment are calculated for these features.

### 3.4 Temporal Modeling of Features

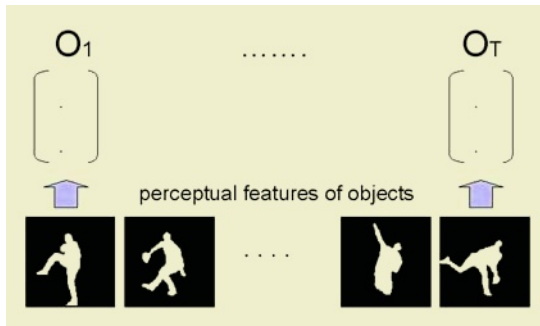
HMM (Hidden Markov Model) has been successfully utilized in speech recognition area and was applied to some image and video applications such as gesture recognition, similarity-based image indexing, video shot detection, and so on [13]. This stochastic automata was exploited in this paper for modeling semantic feature by fusing perceptual features.

In our approach, we regard VO of a frame as a rigid-body object at the specific time instant, gather a series of feature vectors of these rigid bodies through video sequence, and characterize a semantic model by differentiating their temporal variations between adjacent frames.

After obtaining perceptual features described previously for each frame, a feature vector  $\vec{O}_t$ , where t is a frame number, is formed by serializing those features for each frame. An observation sequence  $\vec{O} = \vec{o}_1, \dots, \vec{o}_T$ , where T is total number of frames produced by previous abstraction step, is built by arranging those feature vectors through the all frames. (See Fig. 4) The feature sequence calculated on each video clip for the particular semantic feature are used to train individual HMM model. Due to the sequential nature of video in time domain, the semantic models are designed into left-right HMM in general.



**Fig. 3.** Motion feature extraction. (a-b) Adjacent frames of bowling sequence after abstraction. (c) Motion vectors between (a) and (b). (d) Motion fields after applying object mask

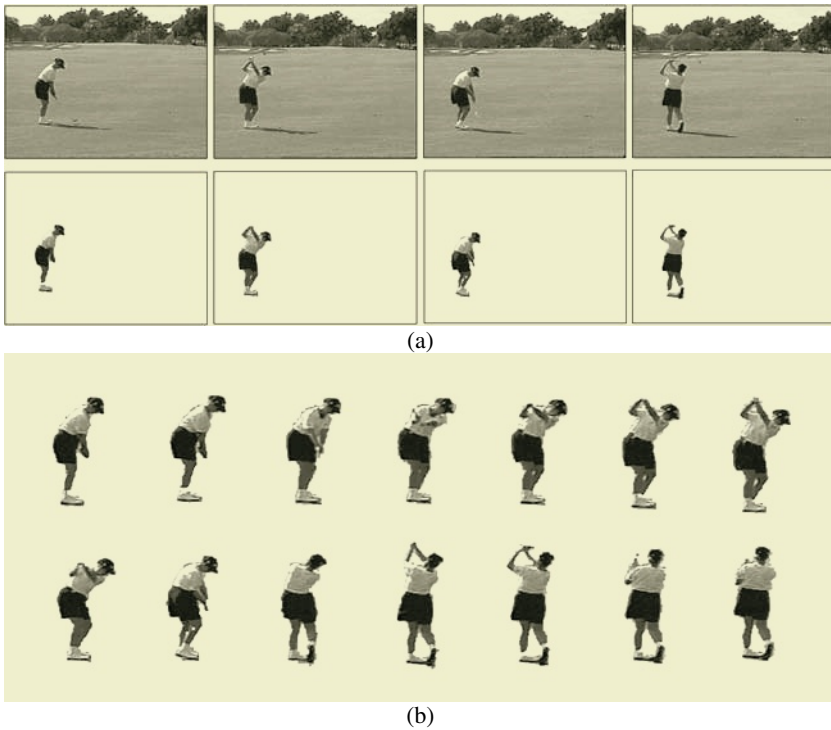


**Fig. 4.** Creation of observation sequence for hidden Markov model (HMM)

## 4 Results

Video sequence for query consists three types of semantic events, downhill skiing, golf swing and ski jump. Each clip was segmented by object tracking algorithm which was explained in Section 2. After segmentation step, extracted VO sequence was abstracted by clustering analysis that was described in [8]. Fig. 5(a) and (b) shows the results of these steps respectively.





**Fig. 5.** Results of VO Extraction(a) and Abstraction(b) of a golf swing clip

We applied our approach to modeling semantic events based on temporal variations of human behaviors within a video sequence.

In training stages, five different types of events (downhill skiing, golf swing, pitching, bowling, and ski jump) were modeled, which are based on temporal changes of object body in sequential manner. To make the HMM more reliable, we extracted the VOs with manual interactions in some video clips such as pitching and bowling, both of which have complex and fast moving backgrounds. Sometimes, it is useful in training stage so that it may preserve the homogeneities and prevent the potential data corruption caused by segmentation errors.

For moment features, the object boundary of each frame was divided into ten blocks. First moment, second and third central moments were calculated for each boundary segment block. Therefore, 30 features were used for each frame. Motion features were also divided into two equally spaced domains in  $x$ ,  $y$  directions and formed four regions. First moment of motion vectors of  $x$  and  $y$  were explored as statistical information on these features and formed 8 features for motion. For UFF, 76 coefficients were considered. Therefore, by combining all the features above, observation vector  $\vec{O}_t$  is created with total 114 elements. Seven states were used for downhill skiing and ten for the others.

Table 1 shows the experimental results of the proposed approach. In the experiment, to begin with, 76 UFF features was tested first to classify the event among 5

different categories. As a result, one golf clip, two ski jump clips and one bowling clips were misclassified. The results means that fusion of multiple features which can represent various aspect of object behavior, is needed for more reliable detection of high level features. In the last column of Table 1, we combined all 3 types of features and demonstrated that the detection ratio was improved. The algorithm missed only one ski jump clip while all the other 32 clips correctly categorized.

**Table 1.** Classification Results: Two right-most columns represents the number of clips which were correctly classified

	Training sets	Query sets	UFF	UFF Moments Motions
Downhill	24	6	6	6
Golf swing	19	4	3	4
Ski jump	15	5	3	4
Pitching	20	9	9	9
Bowling	14	9	8	9

## 5 Conclusion

In this paper, an integrated approach has been proposed for object-based video indexing and retrieval scheme. The abstraction algorithm applied in the proposed approach provides good compression of redundant information from video shots and yields reliable feature data as input to high level feature modeling. Semantic feature modeling based on temporal variation of perceptual features of VO was also described. The proposed algorithm was tested by modeling semantic events based on human behaviors. The experiments have shown promising results and good prospect in developing advanced level feature indexing systems.

For future works, we will make more experiments on features for semantic modeling and develop feature refinement algorithms by exploring feature selection methods.

## References

1. Brunelli, R., Mich, O., Modena, C.M.: A Survey on the Automatic Indexing of Video Data. *Journal of Visual Communication and Image Representation*, 10 (1999) 78-112.
2. Lu, G.: Techniques and Data Structures for Efficient Multimedia Retrieval Based on Similarity. *IEEE Transactions on Multimedia*, 4 (2002) 372-384
3. Zhong, D., Chang, S.: An Integrated Approach of Content-Based Video Object Segmentation and Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology* 9 (1999) 1259-1268
4. Naphade, M. R., Huang, T.S.: A Probabilistic Framework for Semantic Video Indexing, Filtering, and Retrieval. *IEEE Transactions on Multimedia*, 3 (2001) 141-151

5. Haering, N., Qian, R.J., Sezan, M.I.: A Semantic Event-Detection Approach and Its Application to Detecting Hunts in Wildlife Video. *IEEE Transactions on Circuits and Systems for Video Technology* 10 (2000) 857-867
6. Pfeiffer, S. et al: Abstracting Digital Movies Automatically. *Journal of Visual Communication and Image representation* 7 (1996) 345-353
7. Kim, C., Hwang, J.: Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Transactions on Circuits and Systems for Video Technology* 12 (2002) 122-129
8. Kim, C., Hwang, J.: Object-based Video Abstraction for Video Surveillance Systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 12 (2002) 1128-1138
9. Jain, A.K.: *Fundamentals of Digital Image Processing*, Prentice Hall (1989) 344-346.
10. Rauber, T.W., Steiger-Garcia, A.S.: 2-D Form Descriptors Based on a Normalized Parametric Polar Transform (UNL Transform). *MVA'92—IAPR Workshop on Machine Vision Applications*, Japan (1992)
11. Mehtre, B.M., Kankanhalli, M.S., Lee, W.F.: Shape Measures for Content Based Image Retrieval: A Comparison. *Information Processing & Management* 33 (1997) 319-337
12. Bierling, M.: Displacement estimation by hierarchical block matching. *SPIE Visual Commun. Image Processing, VCIP'88*, Cambridge, MA 1001 (1988) 942-951
13. Lin, H.-C., Wang, L.-L., Yang, S.-N.: Color Image Retrieval Based on Hidden Markov Models. *IEEE Transactions on Image Processing* 6 (1997) 332-339

# Video Retrieval Using an EDL-Based Timeline

José San Pedro<sup>1</sup>, Nicolas Denis<sup>2</sup>, and Sergio Domínguez<sup>1</sup>

<sup>1</sup> UPM - DISAM, Universidad Politécnica de Madrid

<http://www.disam.upm.es/vision>

<sup>2</sup> Omnividea Multimedia - Madrid

<http://www.omnividea.com>

**Abstract.** In creating a new multimedia asset, specially a video, some decisions have to be made: a selection of the portions of the original footage that might be included, how to order them, how to crop each portion in order to reach the desired length and how to stitch all these pieces together. All these decisions constitute the core of the so called Editing Decision List, where all these actions are stored for the record. In this paper the authors show that the list of editing decisions can be used as the basis for indexing and retrieving videos from a database; more specifically, we show that a timeline created from the EDL is a valid and sufficient descriptor for identifying a video among a huge population, assuming a minimum duration. We demonstrate, as well, that this descriptor has a very good behavior in terms of robustness given different bit and frame rates, sizes and re-encoding processes. Indexing and retrieval using this descriptor is tested in a IPMP application for TV broadcasting.

## 1 Introduction

Research community is devoting a lot of efforts to develop new paradigms for querying visual databases (typically still and moving pictures) in order to retrieve this visual material using languages which are consistent with the contents to be retrieved. One of the most promising lines of work is the QBE (Query By Example) paradigm, where a piece of visual information is used to launch the query; after that, automatic feature extraction and processing is performed to extract visual cues that will be compared with the database contents. One of the most critical aspects to be taken into account in developing such systems is the kind of visual cues that must be extracted in order to achieve effective indexing and retrieval; of course, this decision should be guided by the definition of similarity within the system.

The question of the similarity definition is linked with the goal of the database system: very often this goal is to resemble as much as possible the selection process that a human operator would carry out in developing the same task as the database manager. However, there are other type of tasks where this human resemblance lays in a second plane, being the main objective to identify or detect the query object.

Being the case of a system for identity detection, different possibilities can be introduced: requirements can ask for a one-to-one identity detection, or part-to-one (where identity must be detected even when just a part of the database object is presented as the query). Typically, in a part-to-one application objects in the database comprise mas-

sive information, like books (find a sentence within a volume) or, and that's the case described in this paper, videos (find a video containing this segment).

In this paper the authors introduce a database system that works under these constraints: QBE paradigm, identity detection for the 100% of the cases and the necessity of performing part-to-one identifications, working with a video database. Additionally, the descriptors we introduce are robust under different video source transformations, like bitrate reduction, frame rate reduction, spatial reduction and re-coding, conditions that perfectly resemble a real multiband, multiplatform distribution environment.

Section 2 summarizes the state of the art in video indexing and retrieval, Section 3 analyzes the proposed descriptor, Section 4 describes algorithms for comparing such descriptors and Section 5 describes the test application built and the results obtained. Finally, in Section 6 a brief set of conclusions are presented.

## 2 Related Work

Video indexing is a problem that has been widely addressed by the image processing research community. Traditionally, the main focus of attention in the QBE field has been the definition of a similarity rating procedure, assuming that the concept of similarity is centered on resembling the human perception. Being this the objective, research lines have mainly followed the path of searching among visual cues in order to find those with stronger power in describing perceptual similarity. Henceforth, most of the work has dealt with such visual cues, mainly three: color, shape and motion [1].

Once these visual cues are extracted, the following step is synthesizing a similarity function that gathers all this information and generates a unique similarity rating. Depending on the application, type of query and other factors, this similarity function can be generated in many different ways; the simplest one could be applying euclidean distance to each couple of feature vectors, while the most complex one could be generated trying to resemble human perception, and even dynamically adjusted by means of mechanisms as relevance feedback [2][3].

Given the fact that a complete video usually contains diverse kinds of footage, in terms of color, shape and motion variety, the common approach is to split the clip in homogeneous segments, having homogeneous values of each of these features. The way to reach such segmentation is to find and consider each shot separately, using different algorithms focused in different kinds of transitions. Usually, these transitions are grouped in: hard cuts, fades, dissolves and wipes [4].

However, few work can be found on managing huge databases for footage identification, leaving apart the concept of similarity and replacing it with the concept of identity. Normally, it can be stated that visual cues have a good discriminant power in order to achieve this goal, but carrying excessive computational costs, both in terms of extraction, storing and handling, since descriptors trends to be massive even for short clips. Usually, finding the identity comprises less information than finding similarity, and therefore reduced feature sets can be used for achieving the same result.

## 3 Video Descriptor

The process of querying a database depends on how the database must be explored in order to build the result set [5]. There are some cases where the query and the corre-

sponding exploration of the database can be set to return just one element, the so called *exact match* or *point query*. This could be the case of a database system working on the identification of the query item, e.g. a face detection application for access control. This is the kind of the application framework that have inspired the work presented herein. Therefore, our first objective is to find a descriptor or descriptors with the maximum robustness in locating identities within a huge database.

### Extended Analysis of Requirements for Retrieval

**Retrieved Set.** The query must generate either a single match in the database, the case when the query object already exist in the database, or none, the case when it doesn't. This means that result must be submitted to a thresholding process, where the similarity measure found between the query object and the retrieved asset is validated to determine if the match is real.

**Time Restrictions.** This requirement has a double implication: computation time of the descriptor and matching time on large databases. The computation of the descriptor must rely on video content, so given the huge amount of data that a video comprises, this is a very time consuming task. On the other hand, seeking the database is a problem whose complexity grows exponentially as the population increases. In our case, the applications must work in real time so our descriptor must be kept as simple as possible taking into account that the objective is not to *reach an interpretation* of the video content, but to *determine whether it is or is not in the database*.

**Matching Restrictions.** As it was stated in the introduction section, it is necessary to be able to match parts of videos. Considering that this part can belong to any moment in the original video, the number of possible matching combinations grows exponentially.

**Tolerance Requirements.** Videos can be extracted from different types of channels or compressed using different codecs or bitrate values. Hence, the descriptor must be flexible enough to allow matching of non-identical videos. The implications concern mainly to the descriptor comparison algorithm.

**Description by Detected EDL.** Videos are complex elements, comprising a huge amount of data which can be structured at different levels of representation, ranging from pixels to segments. Therefore, the task of comparing a pair of such elements is correspondingly complex, since comparison can be performed on the basis of any level of representation [6].

In this paper the authors introduce the timestamp of edition effects as a sufficient and robust descriptor that we have called *detected EDL*, or detected Edit Decision List<sup>1</sup>. By edition effect we mean any transition between shots or any effect that is considered relevant enough to be included in the timeline. Similar algorithms are used for different identification tasks, as the discid algorithm used by the freedb database to identify music albums.

---

<sup>1</sup> Edit Decision List is a record explaining which pieces of original footage have been used and how they have been stitched together to build a clip.

**Edition Effects.** It is widely accepted that there are three main edition effects that sum up the 95% of the grand total: hard cuts, fades and dissolves. Hard cuts are relatively easy to find at a relatively low computational cost. Fades and dissolves require specific detection algorithms with low performance, in terms of a bad balance between false positives and missed effects, in addition to high computational. This poor performance along with the fact that around 90% of shot transitions are hard-cuts, as it is stated in [4], made us choose only hard cuts to build up the detected EDL timeline.

To detect hard cuts, the algorithm presented in [7] is used. It computes the difference between consecutive frames color histograms obtaining a temporal series of values that represent color change from a frame to the following along all the video. Detecting a hard-cut is as easy as detecting local maxima in this series.

**Analysis Flaws.** The CHD algorithm introduced above has a slight error rate that needs to be properly handled by the application. The query sample provided will not be normally extracted from the same source as the object stored in the database, as they represent different instances of the same original material; the differences can be produced due to variations on different media types, coding algorithms, bitrates, framerates, interlacing and so on. Therefore, it is possible that the detected EDL varies for different instances of a video due to precision issues of the detection algorithm (causing a number of non-corresponding effects between the two timelines):

- An effect was detected in the database object but not in the query object: produced by a false positive in the first or by a missed effect in the second.
- An effect was not detected in the database object but was in the query object.
- An effect was detected in both instances but with not exactly at the same video time: this can be produced by recordings at different bitrate or framerate, or recordings using different algorithms.

**Descriptor Implementation.** Given a video clip  $V$ , let its Detected EDL (DEDL) be  $T_V = \{t_1, t_2, t_3, \dots, t_N\}$ , where  $t_i$  stands for the amount of time units between hard cuts  $H_{i-1}$  and  $H_i$ . This information will be stored in the database for featuring a video, and extracted from a query object as search key. Notice that the timeline store differences between consecutive elements. Being  $h_i$  the time at which the  $H_i$  hard cut is detected, the descriptor is built up by storing  $t_i = h_i - h_{i-1}$ , as stated before, defining  $h_0 = 0$ , the beginning of the clip. This definition brings several advantages:

*Storage Size.* Time stamps need to be stored with subsecond precision. If absolute time values are stored, the representation of time stamps at the end of long videos will increase dramatically storage size (and thus decrease performance). By storing differences, values tend to be smaller.

*Ease of Comparison.* When comparing timelines, it is common to come into cases as the one show in figure 1. In it, the maximum similarity value is not found in the first position of the timeline. If dealing with absolute time stamps, a reference must be chosen and all time stamps need to be recomputed. In our timeline model there is no need to modify time stamps because the reference is always the previous value: performance is greatly improved.

In the next sections, a deep analysis of the use of this descriptor as a tool for video comparison is provided.

## 4 Timeline Comparison

Our application will normally deal with large timelines (coming from an input video broadcast), and will compare it with a big set of much smaller timelines (from the copyrighted videos database). In this scenario, comparing a very short timeline to a much larger one could provide a similarity value close to 1, if the shorter timeline is a part of the larger one, as shown in figure 1.



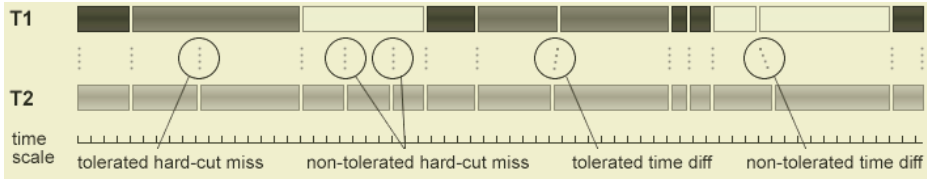
**Fig. 1.** Comparing different sized timelines. If the shorter is part of the larger, a high similarity value is produced.

Additional constraints arise because of accuracy problems in hard-cut detection. Two different copies of the same material may produce a different result when applied to the hard-cut detection algorithm. Two kinds of tolerance are introduced:

- Time deviation tolerance: to deal with slight differences in position of detected hard cut caused by deviations in the stream flow or by the hard-cut detector.
- Hard-cut missing tolerance: to deal with missed and false detections introduced by the video analyzer. The importance of missing one time stamp, or having an extra one, should not be considered in a local extent but globally.

Most of the previous work was created to find similarity between character strings. Considering each element of the timeline a letter in an appropriate alphabet such methods can be applied in the general timeline case. These classical methods are mainly based in measuring distance between corresponding elements in the timelines (hamming distance) or finding the longest subsequence of elements that is included in both timelines (longest common subsequence). This kind of algorithms get poor results when the sequences being are likely to include missing elements. A better idea is to provide the option of applying transformations to the sequences. The “edit distance method” performs the measurement in a very different fashion using the idea of transformations. Three basic transformations are provided: insertion, deletion and change. Each transformation has a penalty value (e.g. hamming distance, for changes). By using these transformations more robustness in the presence of missing hard-cuts is achieved. As the requirements for each application made comparison a very specific problem, two new methods have been created to improve results obtained with classical methods.





**Fig. 2.** Similar time addition method. Similar hard-cuts are shown in black, tolerated hard-cuts in grey, and different hard-cuts in white. Here the first tolerated hard-cut takes into account an “hard-cut missing” tolerance, and the second one takes into account a “temporal tolerance”.



**Fig. 3.** Morphing distance based method. To be able to transform the first timeline,  $T1$ , into the second,  $T2$ , we need to move a hard-cut (1), merge two shots (2), and split a shot into two (3).

*Similar Time Addition.* When two sequences have a high similarity value then they have a large number of equally placed hard-cuts. An intuitive way to measure the similarity value between two sequences at a certain position, is to measure the total amount of similar time, which is the sum of all the shot duration between hard-cuts equally placed in both. A tolerance value, which can be computed as a function of the time since the last hard-cut, is set to deal with slight timeline differences (see figure 2).

A similarity coefficient can be obtained by computing the duration of matched shots. A penalty value,  $0 \leq \omega_{cost} \leq 1$ , can be applied to the extra duration obtained thanks to tolerance values. Then we can define a coefficient similarity as:

$$Similarity = \frac{\text{matched shot time} + \omega_{cost} \text{ tolerated shot time}}{\text{Clip length}}$$

*Morphing Distance.* This distance represents the minimum amount of transformations needed to transform a sequence into another. If two sequences are similar, fewer transformations are needed, and the “morphing distance” will be smaller.

Firstly, it is necessary to define the allowed transformations in this method (see figure 3). Each video is divided into shots, each of them defined as the segment between two consecutive hard-cuts. The first transformation is splitting a segment in two. This transformation allows to deal with hard-cut detection misses and it is equivalent to introducing a new hard-cut. The second transformation is merging two segments together. This is the opposite case and allows to deal with hard-cut false detection. The last transformation is shifting the position of a hard-cut. It allows to deal with small differences of hard-cut timestamp detection. Each transformation has an associated cost. The cost of the operations can be defined depending on the application. In our case, the merge operation is set to 1, the split operation is set also to 1 and, finally, the cost of a move operation will depend on the distance moved. For this last operation, a tolerance value can be set so a hard-cut can be freely moved within that environment.

The similarity value between two timelines,  $T1$  and  $T2$ , with lengths  $|T1|$  and  $|T2|$  respectively, is defined as:

$$Similarity = \frac{\text{Cost of all the transformations}}{\min(|T1|, |T2|)}$$

An interesting improvement to this method is to assign a variable cost to “merge” and “split” operations depending on their neighborhood. When the neighbor hard-cuts are well matched, the cost is reduced, thus taking into account that it is possible to detect some false or missing hard-cuts.

## 5 Results

### 5.1 Descriptor Generation

Generation of the descriptor can be performed much faster than real time. Computation time depends mainly on spatial and temporal resolution. Temporal resolution (frame rate) has a direct linear impact on the final time because it defines the number of frames that will be processed by the algorithm. Spatial resolution affects also linearly to computation time (as it grows, processing each frame takes more time). It is important to note that hard-cut detection accuracy is not severely affected by a significant quality decrease, allowing to perform this stage very fast.

### 5.2 Matching Accuracy

In this section, a comparison of the algorithms discussed in section 4 is provided. The goal of the first set of tests was to find out the best algorithm for comparing sequences. Though the similarity value found with all of them is very good, the morphing distance algorithm was chosen because it is able not only to get high similarity values with similar videos but also it is able to get low values for different videos. A brief summary of the results is shown in table 1. To test matching accuracy, several versions of the same video have been generated, changing spatial and temporal resolutions as well as quality. These have been fed to the morphing algorithm. Results are shown in table 2.

Similarity values obtained by the morphing algorithm are quite good. Reducing both bitrate and spatial resolution in the incoming video, in order to save the descriptor computation time, does not produce a significant decrease of similarity, which remains

**Table 1.** Timeline comparison algorithms performance. When comparing similar timelines, all algorithms find high similarity. For different timelines, morphing shows the best behavior.

Algorithm	Similarity for variations of same video	Similarity for a different video
Hamming	98.51%	27.48%
Edit Distance	99.19%	23.54%
Similar Time	98.73%	19.36%
Morphing	99.95%	12.09%

**Table 2.** Morphing algorithm matching accuracy, Database Size: 30 videos (mean video duration  $\simeq$  3min30sec).

Video	Frame Rate	Bit Rate	Resolution	Duration (sec)	Similarity
$V_1$	25	2000	352x288	272	100%
$V_1$	25	1000	200x150	272	95.35%
$V_1$	12	2000	352x288	272	92.19%
$V_1$ (Segment)	25	2000	352x288	20	96.54%
$V_1$ (Different Encoder)	25	2000	352x288	272	97.72%
$V_2$	25	2000	352x288	272	12.09%
$V_1$ in large incoming Stream	25	250	352x288	1844	90.02%

above 90%. The same applies to the frame rate reduction. It is important to notice the great similarity value obtained for a segment of the original video (entry number 2). It is also important to remark the low similarity value found for a different video.

## 6 Conclusion

This paper introduces a video descriptor and a comparison function, both easy to extract and robust, that allows to perform fast identity retrieval from a database, in contrast to similarity based retrieval. The proposed retrieval system obtains good identity values even when querying variations of the original content (lower frame rate or quality). It can be stored in a database, along with other relevant information of the video, using the MPEG-21 standard.

## References

1. del Bimbo, A.: Visual Information Retrieval. Morgan Kauffman Publishers, Inc. (1999)
2. Jolion, J.M.: Feature Similarity. In: Principles of Visual Information Retrieval. Springer (1999)
3. Rui, Y., Huang, T.S.: Relevance Feedback Techniques in Image Retrieval. In: Principles of Visual Information Retrieval. Springer (1999)
4. Lienhart, R.: Comparison of Automatic Shot Boundary Detection Algorithms. In: Image and Video Processing VII, SPIE (1999)
5. Brown, L., Gruenwald, L.: Tree-Based Indexes for Image Data. Journal of Visual Communication and Image Representation **9** (1998) 300–313
6. Smith, M.A., Chen, T.: Image and Video Indexing and Retrieval. In: Handbook of Image and Video Processing. Academic Press (2000)
7. Boreczky, J.S., Rowe, L.A.: Comparison of Video Shot Boundary Detection Techniques. In: Storage and Retrieval for Still Image and Video Databases IV, SPIE (1996)
8. Y, S.: A new video segmentation method using variable interval frame differencing for digital video library applications. Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access (2003)
9. S, P., M, M., B, T.: Temporal video segmentation and classification of edit effects. IMAGE AND VISION COMPUTING **21** (2003) 1097–1106

## Part IV

# Image and Video Coding

# A New Secret Sharing Scheme for Images Based on Additive 2-Dimensional Cellular Automata

Gonzalo Álvarez Marañón<sup>1</sup>,  
Luis Hernández Encinas<sup>1</sup>, and Ángel Martín del Rey<sup>2</sup>

<sup>1</sup> Instituto de Física Aplicada, CSIC, C/ Serrano 144, 28006-Madrid, Spain  
{gonzalo,luis}@iec.csic.es

<sup>2</sup> EPS, Universidad de Salamanca, C/ Sto. Tomás s/n, 05003-Ávila, Spain  
delrey@usal.es

**Abstract.** A new secret color image sharing scheme based on two-dimensional memory cellular automata, is proposed. Such protocol is of a  $(n, n)$ -threshold scheme where the secret image to be shared is considered as one of the initial configurations of the cellular automata. The original idea is to study how a reversible model of computation permits to compute the shares and then using the reverse computation in order to recover the original image. The scheme is proved to be perfect and ideal, and resistant to the most important attacks such as statistical attacks.

**Keywords:** Cellular automata, Cryptography, Image processing, Secret sharing.

## 1 Introduction

*Secret sharing schemes* are cryptographic procedures to share a secret among a set of participants in such a way that only some qualified subsets of these participants can recover the secret. Such schemes were independently introduced by Shamir ([13]) and Blakley ([3]) and their original motivation was to safeguard cryptographic keys from loss. Currently, there are many applications in different areas such as access control, opening safety deposit boxes, etc.

The basic example of secret sharing scheme is the  $(k, n)$ -*threshold scheme*, where  $k$  and  $n$  are integer numbers such that  $1 \leq k \leq n$ . The structure of this scheme is as follows: There exists a mutually trusted party (or a dealer) which computes  $n$  secret shares from an initial secret and securely distributes them into  $n$  participants in such a way that any  $k$  or more of these participants who pool their shares may easily recover the original secret, but any group knowing only  $k - 1$  or fewer shares are unable to recover the secret. Shamir's scheme, which is based on polynomial interpolation, and Blakley's scheme, based on the intersection of affine hyperplanes, are examples of  $(k, n)$ -threshold schemes.

A  $(k, n)$ -threshold scheme is *perfect* if the knowledge of any  $k - 1$  or fewer shares provide no information about the original secret. Moreover, a  $(k, n)$ -threshold scheme is *ideal* if the size of every share is equal to the size of the shared secret. For a more detailed description of these schemes we refer the

reader to [10], [14], and [15]. The first scheme proposed to share images was based on visual threshold schemes  $k$  of  $n$  (see [12]) and is called visual cryptography. This visual scheme is perfect but not ideal since the size of the shared images is bigger than the original one. Moreover, in the visual schemes, there is a great contrast loss between the secret image and the recovered one.

Furthermore, visual secret sharing schemes have been proposed for several applications. For example, in [5] a method for intellectual property protection of grey level images is presented, and a scheme to share multiple secrets by using digital images is proposed in [18]. A recent secret sharing scheme for grey-level images which elaborates shares of smaller size than the original image is presented in [16]. It is based on Shamir scheme, but the grey value range is limited to only 250 levels. Finally, in [6] another algorithm for sharing images, not based on the visual cryptography, has been proposed.

In this work we use a particular type of two-dimensional delay discrete dynamical systems, called two-dimensional memory cellular automata, in order to share a secret color image. The main features of the proposed scheme are two: (i) The shares obtained for each participant have the same size as the secret image, and (ii) The recovered image is exactly the same as the original one, that is, there is no loss of resolution. The use of cellular automata to design cryptosystems goes back to the mid-80s ([20]). In the recent years many cryptographic protocols have been proposed (see, for example, [2], [4], [7], [8], [9], [11], [19]).

The remainder of the paper is organized as follows: In Section 2, the basic definitions about memory cellular automata are introduced; in Section 3 the secret sharing scheme based on memory cellular automata is presented, and its security analysis is given in Section 4. The conclusions are shown in Section 5.

## 2 Two-Dimensional Memory Cellular Automata

*Two-dimensional finite boolean cellular automata* (2D-CA for short) are discrete dynamical systems formed by a finite two-dimensional array of  $r \times s$  identical objects called cells, in such a way that each of them can assume a state, which is an element of a finite set,  $S = \mathbb{Z}_2$ . The  $(i, j)$ -th cell is denoted by  $\langle i, j \rangle$ , and the state of this cell at time  $t$  is given by  $a_{ij}^{(t)}$ . The 2D-CA evolves deterministically in discrete time steps, changing the states of all cells according to a local transition function. The updated state of each cell depends on the variables of the local transition function, which are the previous states of a set of cells, including the cell itself, and constitutes its neighborhood. In this work extended Moore neighborhoods are considered; that is, the eight nearest cells to the cell  $\langle i, j \rangle$  and itself are considered. This neighborhood will be denoted by  $V_{ij}$ . As a consequence, the local transition function takes the following form:

$$a_{ij}^{(t+1)} = f \left( V_{ij}^{(t)} \right), \quad 0 \leq i \leq r-1, 0 \leq j \leq s-1,$$

where  $V_{ij}^{(t)}$  stands for the states of the neighbor cells of  $\langle i, j \rangle$  at time  $t$ :

$$V_{ij}^{(t)} = \left\{ a_{i-1, j-1}^{(t)}, a_{i-1, j}^{(t)}, a_{i-1, j+1}^{(t)}, a_{i, j-1}^{(t)}, a_{ij}^{(t)}, a_{i, j+1}^{(t)}, a_{i+1, j-1}^{(t)}, a_{i+1, j}^{(t)}, a_{i+1, j+1}^{(t)} \right\}.$$

The matrix  $C^{(t)} = \left( a_{ij}^{(t)} \right)$  is called the configuration at time  $t$  of the 2D-CA, and  $C^{(0)}$  is the initial configuration of the CA. Moreover, the sequence  $\{C^{(t)}\}_{0 \leq t \leq k}$  is called the evolution of order  $k$  of the 2D-CA, and  $\mathcal{C}$  is the set of all possible configurations of the 2D-CA; consequently  $|\mathcal{C}| = 2^{r \cdot s}$ . As the number of cells of the 2D-CA is finite, boundary conditions must be considered in order to assure the well-defined dynamics of the cellular automaton. In this work periodic boundary conditions are taken: if  $i \equiv u \pmod{r}$ , and  $j \equiv v \pmod{s}$ , then  $a_{ij}^{(t)} = a_{uv}^{(t)}$ . The global function of the 2D-CA is a linear transformation,  $\Phi: \mathcal{C} \rightarrow \mathcal{C}$ , that yields the configuration at the next time step during the evolution of the 2D-CA, that is,  $C^{(t+1)} = \Phi(C^{(t)})$ . If  $\Phi$  is bijective then there exists another cellular automaton, called its inverse, with global function  $\Phi^{-1}$ . When such inverse 2D cellular automaton exists, the 2D-CA is called reversible and the backward evolution is possible ([17]).

Let us consider the set of 2D-CA whose local transition function takes the following form:

$$a_{ij}^{(t+1)} = \sum_{\alpha, \beta \in \{-1, 0, 1\}} \lambda_{\alpha, \beta} a_{i+\alpha, j+\beta}^{(t)} \pmod{2},$$

where  $0 \leq i \leq r-1$ ,  $0 \leq j \leq s-1$ , and  $\lambda_{\alpha, \beta} \in \mathbb{Z}_2$ . These are called 2D linear cellular automata (2D-LCA for short). As there are 9 cells in the extended Moore neighborhood, then there exist  $2^9 = 512$  different 2D-LCAs, and every one of them can be conveniently specified by a decimal integer called rule number:  $\omega$ , which is defined as follows:

$$\begin{aligned} \omega = & \lambda_{-1, -1} 2^8 + \lambda_{-1, 0} 2^7 + \lambda_{-1, 1} 2^6 + \lambda_{0, -1} 2^5 + \lambda_{0, 0} 2^4 \\ & + \lambda_{0, 1} 2^3 + \lambda_{1, -1} 2^2 + \lambda_{1, 0} 2^1 + \lambda_{1, 1} 2^0, \end{aligned}$$

where  $0 \leq \omega \leq 2^9 - 1$ . Note that the 2D-LCA with  $\omega = 16$  stands for the identity.

The standard paradigm for 2D-CA claims that the state of every cell at time  $t+1$  depends on the state of some cells (its neighborhood) at time  $t$ . Nevertheless, one can consider 2D-CA for which the state of every cell at time  $t+1$  not only depends on the states of some cells at time  $t$  but also on the states of (possibly) another different groups of cells at times  $t-1$ ,  $t-2$ , etc. This is the basic idea of two-dimensional memory cellular automata ([1]), 2D-MCA for short. In this work, we consider a particular type of 2D-MCA called  $k$ -th order linear MCA (2D-LMCA for short) whose local transition functions takes the following form:

$$a_{ij}^{(t+1)} = \sum_{l=1}^k f_l \left( V_{ij}^{(t+1-l)} \right) \pmod{2}, \quad (1)$$

with  $0 \leq i \leq r-1$ ,  $0 \leq j \leq s-1$ , and  $f_l$  is the local transition function of a particular 2D-LCA. In this case, the configurations  $C^{(0)}, \dots, C^{(k-1)}$  are called initial configurations of the  $k$ -th order 2D-LMCA.

It is a well-known fact (see [1]) that if  $f_k \left( V_i^{(t-k+1)} \right) = a_i^{(t-k+1)}$ , then the 2D-LMCA given by (1) is a reversible CA, whose inverse 2D-CA is another  $k$ -th order 2D-LMCA with local transition function:

$$a_{ij}^{(t+1)} = \sum_{m=0}^{k-2} f_{k-m-1} \left( V_{ij}^{(t-m)} \right) + a_{ij}^{(t-k+1)} \pmod{2},$$

for  $0 \leq i \leq r - 1, 0 \leq j \leq s - 1$ .

### 3 The Secret Sharing Scheme for Images

In this section we propose a new secret sharing scheme. It consists of a  $(n, n)$ -threshold scheme such that the color image to be shared,  $I$ , is one of the initial configurations of a reversible  $n$ -th order 2D-LMCA, specifically  $C^{(n-1)}$ . The remainder initial configurations,  $C^{(0)}, \dots, C^{(n-2)}$ , are  $n - 1$  random matrices of the same size as  $I$ . The shares to be distributed among the  $n$  participants are  $n$  consecutive configurations of the evolution of the LMCA.

An arbitrary image  $I$  defined by  $a \times b$  pixels,  $p_{ij}$ , with  $0 \leq i \leq a - 1, 0 \leq j \leq b - 1$ , and  $c$  colors, can be considered as a configuration, for example  $C^{(t)}$ , of a 2D boolean cellular automata with  $r \times s$  cells as follows:

1. If  $I$  is a binary image, *i.e.*  $c = 2$ , then  $a_{ij}^{(t)} = 0$  if the pixel  $p_{ij}$  is black, and  $a_{ij}^{(t)} = 1$  if the pixel  $p_{ij}$  is white. As a consequence, in this case  $r = a$  and  $s = b$ .
2. If  $I$  is a grey-level image, then  $c = 2^8$  and the RGB code of each pixel  $p_{ij}$  can be defined by eight bits. Hence,  $C^{(t)}$  is an  $a \times (8 \cdot b)$  boolean matrix, that is,  $r = a$  and  $s = 8 \cdot b$ . A similar configuration appears if the image is defined by 256 colors.
3. Finally, if  $I$  is a color image defined by  $c = 2^{24}$  colors, then each pixel is given by 24 bits. As a consequence  $C^{(t)}$  is an  $a \times (24 \cdot b)$  boolean matrix, and, obviously,  $r = a$  and  $s = 24 \cdot b$ .

#### 3.1 Structure of the Scheme

The procedure to share secret images by means of LMCA is divided into three phases: The setup phase, the sharing phase, and the recovery phase.

##### The Setup Phase

1. The mutually trusted party determines a sequence of  $n$  integer numbers:  $\{\omega_1, \dots, \omega_n\}$ , such that  $\omega_n = 16$ , and the remaining values  $0 \leq \omega_l \leq 511$ , with  $1 \leq l \leq n - 1$ , can be generated by a random bit generator. These  $n$  numbers stand for the rule numbers of the 2D-LCA constituting the 2D-LMCA used.
2. The mutually trusted party constructs the reversible  $n$ -th order 2D-LMCA with local transition function:

$$a_{ij}^{(t+1)} = \sum_{l=1}^n f_{\omega_l} \left( V_{ij}^{(t+1-l)} \right) \pmod{2}, \tag{2}$$

where  $f_{\omega_j}$  is the local transition function of the 2D-LCA with rule number  $\omega_l, 1 \leq l \leq n$ , and  $0 \leq i \leq r - 1, 0 \leq j \leq s - 1$ .



3. The secret image to be shared is considered as the initial configuration  $C^{(n-1)}$ , and the mutually trusted party computes the remaining  $n - 1$  initial configurations:  $C^{(0)}, \dots, C^{(n-2)}$ , by using a random bit generator. These  $n - 1$  configurations must be destroyed after generating the shares.

**The Sharing Phase**

1. The mutually trusted party chooses an integer number  $m$ , such that is public and  $m \geq n$ . For efficiency,  $m$  must be closer to  $n$  in order to reduce the number of iterations of the cellular automaton.
2. Starting from the initial configurations  $C^{(0)}, \dots, C^{(n-1)}$ , the mutually trusted party computes the  $(n + m - 1)$ -th order evolution of the 2D-LMCA:

$$\left\{ C^{(0)}, \dots, C^{(n-1)}, C^{(n)}, \dots, C^{(m)}, \dots, C^{(m+n-1)} \right\}.$$

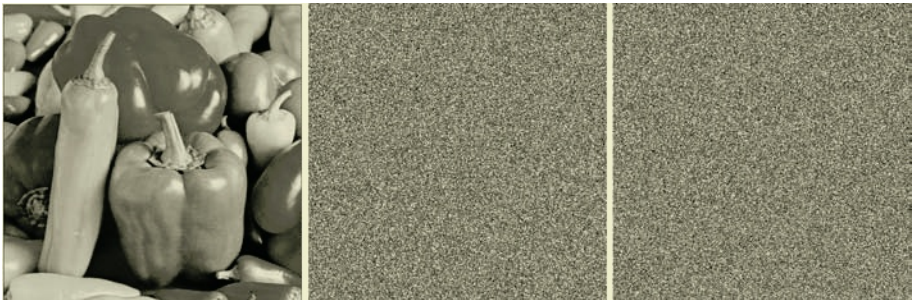
3. The shares to be distributed among the  $n$  participants are the last  $n$  configurations computed:  $S_1 = C^{(m)}, \dots, S_n = C^{(m+n-1)}$ . Note that  $m \geq n$  is considered to avoid overlapping between the initial configurations and the shares.
4. Each participant receives the 3-uplets  $(i, \omega_i, S_i)$ , in order to construct the inverse function of the local transition function given by formula (2).

**The Recovery Phase**

1. To recover the secret,  $C^{(n-1)}$ , all the 3-uplet  $(i, \omega_i, S_i = C^{m+i-1})$ , for  $i = 1, \dots, n$ , are needed.
2. The secret,  $C^{(n-1)}$ , is obtained by taking  $\tilde{C}^{(0)} = C^{(m+n-1)}, \dots, \tilde{C}^{(n-1)} = C^{(m)}$ , and iterating  $m$  times the inverse 2D-LMCA.

**3.2 An Example**

In this section, we present an example for the proposed scheme by using a grey-level image with  $512 \times 512$  pixels (see Fig. 1). For the sake of simplicity, we have computed the protocol for the values  $n = 2 = m$ , and we have obtained two shares (see Fig. 1). The recovered image is exactly the same as the original one.



**Fig. 1.** The original image and its shares.

## 4 Security Analysis

The security of the proposed scheme is considered in this section. In particular, we prove that the scheme is ideal and perfect, and we show that it resists the most important statistical attacks.

### 4.1 The Scheme Is Ideal and Perfect

As the size of every distributed image share is equal to the size of the secret image (recall that both are configurations of the same 2D-LHCA), the proposed scheme is ideal.

Furthermore, the scheme is also perfect since if only one configuration of the form  $C^{(t-i)}$ , with  $0 \leq i \leq n - 1$ , is unknown, say for example  $C^{(t-n+1)}$ , then there is no information about the configuration  $C^{(t+1)}$  as the evolution of the 2D-LMCA is given by the following linear system:

$$a_{ij}^{(t+1)} = b_{ij} + a_{ij}^{(t-n+1)} \pmod{2}, \quad 0 \leq i \leq r - 1, 0 \leq j \leq s - 1,$$

where  $b_{ij} = f_{\omega_1} \left( V_{ij}^{(t)} \right) + \dots + f_{\omega_{n-1}} \left( V_{ij}^{(t-n+2)} \right)$ .

Consequently, it is formed by  $r \cdot s$  equations with  $2r \cdot s$  unknown variables:  $a_{ij}^{(t+1)}, a_{ij}^{(t-n+1)}, 0 \leq i \leq r - 1, 0 \leq j \leq s - 1$ . Hence it can not be solved and, obviously, no information about the configuration  $C^{(t+1)} = \left( a_{ij}^{(t+1)} \right)$ , where  $0 \leq i \leq r - 1, 0 \leq j \leq s - 1$ , is obtained. Note that a similar result holds if the number of unknown configurations is greater than one. As a consequence, for the secret sharing scheme proposed it is impossible to recover the secret image starting from  $n - 1$  (or less) shares.

### 4.2 Statistical Analysis

We have studied statistical analysis in order to prove the confusion and diffusion properties of the proposed scheme, which allows it to strongly resists statistical attacks. This analysis is performed by a test on the histograms and by the correlations of adjacent pixels of the original image and its shares.

The histograms of the original image and the shares given in Fig. 1 are shown in Fig. 2. One can see that the histograms of the shares are fairly uniform and they are significantly different from the histogram of the original image.

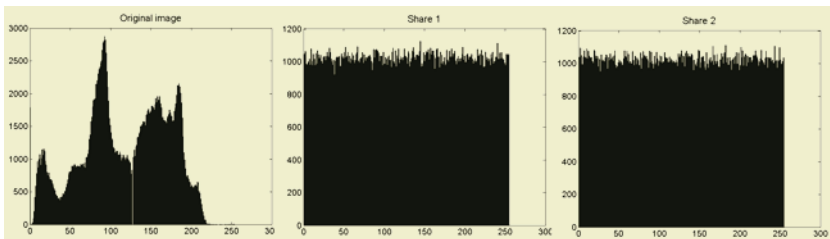
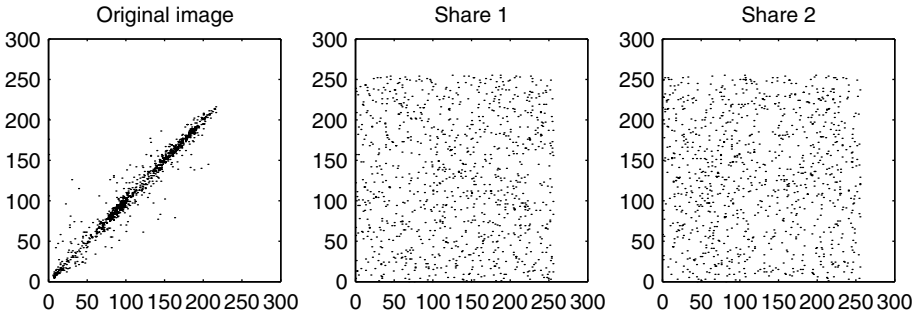


Fig. 2. Histograms of the original image and its shares.

**Table 1.** Correlation coefficients of two adjacent pixels.

	Original Image	Share 1	Share 2
Horizontal	0.9802	0.0399	0.0015
Vertical	0.9840	-0.0240	-0.0394
Diagonal	0.9576	-0.0009	0.0063

**Fig. 3.** Correlation of diagonally adjacent pixels.

To test the correlation between two adjacent pixels in the images, we have randomly selected 1000 pairs of two vertically adjacent pixels, 1000 pairs of two horizontally adjacent pixels, and 1000 pairs of two diagonally adjacent pixels, for the original image as well as for its shares. In each case, we have computed the correlation coefficient of each pair and the results obtained are shown in Table 1. One can see that the correlation coefficients are far apart. For example, in the original image, the correlation coefficient for two horizontally adjacent pixels is 0.9802, which is very near to 1, as it was expected. Nevertheless, in the two shares, these coefficients are 0.0399 and 0.0015, respectively, that is, these are very close to 0.

Finally, Fig. 3 shows the correlation distribution of two diagonally adjacent pixels in the original image and in its shares.

## 5 Conclusions

In this paper we have studied the application of a reversible model of computation, based on two-dimensional reversible linear memory cellular automata, to define a new  $(n, n)$ -threshold scheme for image sharing. We have proved that it is ideal and perfect since the size of the shares to be distributed to the participants and the size of the secret image are equal, and no information about the secret is obtained if  $n - 1$  or less shares are known. Moreover, we have shown that the scheme resists the most important statistical attacks. Now, we are working in extending the previous scheme to more general  $k$  of  $n$  schemes, where  $k < n$ .

## Acknowledgements

This work has been supported by Ministerio de Educación y Ciencia (Spain) under grant SEG2004-02418, and by the Consejería de Educación y Cultura of Junta de Castilla y León (Spain), under grant SA052/03.

## References

1. R. Alonso-Sanz, Reversible cellular automata with memory: two-dimensional patterns from a single seed. *Physica D* **175**(2003), 1–30.
2. G. Álvarez Marañón, A. Hernández Encinas, L. Hernández Encinas, A. Martín del Rey, and G. Rodríguez Sánchez, Graphics cryptography with pseudorandom bit generators and cellular automata, *Lect. Notes Artif. Intell.* **2773** (2003), 1207–1214.
3. G.R. Blakley, Safeguarding cryptographic keys, *AFIPS Conference Proceedings* **48** (1979), 313–317.
4. K. Cattell and J.C. Muzio, An explicit similarity transform between cellular automata and LFSR matrices, *Finite Fields Appl.* **4** (1998), 239–251.
5. C.C. Chang and J.C. Chuang, An image intellectual property protection scheme for gray-level images using visual secret sharing strategy, *Pattern Recogn. Lett.* **23** (2002), 931–941.
6. C. Chang and R. Hwang, Sharing secret images using shadow codebooks, *Inform. Sci.* **111** (1998), 335–345.
7. R. Díaz Len, A. Hernández Encinas, L. Hernández Encinas, S. Hoya White, A. Martín del Rey, G. Rodríguez Sánchez, and I. Visus Ruíz, Wolfram cellular automata and their cryptographic use as pseudorandom bit generators, *Internat. J. Pure Appl. Math.* **4** (2003), 87–103.
8. L. Hernández Encinas, A. Martín del Rey, and A. Hernández Encinas, Encryption of images with 2-dimensional cellular automata, *Proc. of 6-th Multiconference on Systemics, Cybernetics and Informatics*, 2002, 471–476.
9. W. Meier and O. Staffelbach, Analysis of pseudorandom sequences generated by cellular automata, *Lect. Notes Comput. Sci.* **547** (1991), 186–189.
10. A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of applied cryptography*, CRC Press, Boca Raton, FL., 1997.
11. S. Nandi, B.K. Kar, and P.P. Chaudhuri, Theory and applications of cellular automata in cryptography, *IEEE Trans. Comput.* **43** (1994), 1346–1357.
12. M. Naor and A. Shamir, Visual cryptography, *Lect. Notes Comput. Sci.* **950**, (1995), 1–12.
13. A. Shamir, How to share a secret, *Commun. ACM* **22** (1979), 612–613.
14. D.R. Stinson, An explication of secret sharing schemes, *Des. Codes Cryptogr.* **2** (1992), 357–390.
15. D.R. Stinson, *Cryptography Theory and Practice*, Second Edition, CRC Press, Boca Raton, FL., 2002.
16. C. Thien and J. Lin, Secret image sharing, *Computers & Graphics* **26** (2002), 765–770.
17. T. Toffoli and N. Margolus, Invertible cellular automata: A review, *Physica D* **45** (1990), 229–253.
18. C.S. Tsai, C.C. Chang, and T.S. Chen, Sharing multiple secrets in digital images, *J. Syst. Software* **64** (2002), 163–170.
19. M. Tomassini and M. Perrenoud, Cryptography with cellular automata, *Appl. Software Comput.* **1** (2001), 151–160.
20. S. Wolfram, Random sequence generation by cellular automata, *Adv. Appl. Math.* **7** (1986), 123–169.

# A Fast Motion Estimation Algorithm Based on Diamond and Triangle Search Patterns

Yun Cheng<sup>1,2</sup>, Zhiying Wang<sup>1</sup>, Kui Dai<sup>1</sup>, and Jianjun Guo<sup>1</sup>

<sup>1</sup> College of Computer, National University of Defense Technology  
410073 Changsha, China  
{chengyun, daikui}@chiplight.com.cn

<sup>2</sup> Department of Computer, College of Hunan Humanities, Science and Technology  
417000 Loudi, China

**Abstract.** Based on the study of patterns used in many fast algorithms for the block-matching motion estimation (BMME), a new search pattern, TP (Triangle Pattern), was introduced in this paper. TP is a simplified SP (Square Pattern), so it has almost the same performance as SP. By combining TP with DP, a fast BMA (BMME Algorithm), DTS (Diamond-Triangle Search), was also proposed in this paper. DTS well exploits the motion correlation between the adjacent blocks, the directional characteristic of SAD(Sum of Absolute Difference) distribution, and the center-biased characteristic of motion vectors to speed up the BMME. Experimental results show that the proposed DTS algorithm can reduce the computational complexity of the BMME remarkably while incurring little, if any, loss in quality.

## 1 Introduction

Motion estimation and compensation are very important components of video coding. They are used to eliminate the temporal redundancy information between successive frames so as to improve the encoding efficiency greatly. BMA(BMME Algorithm) is a widely used motion estimation algorithm and it was adopted by many video-coding standards such as MPEG-1/2/4, H.261, H.263 and H.264/AVC etc. [1] [2]. The most basic BMA is the full search (FS). Although FS can find the best matching block by exhaustively testing all the candidate blocks within the search window, its computation is too heavy: experimental results demonstrate that the time of the BMME consumed by FS in H.264 is about 60% to 80% of the total. In order to speed up the BMME, many researchers have been working hard for many years and have proposed many kinds of fast BMAs.

Most of the fast BMAs find the best matching block (or point) by using some special search patters. For example, TDLs(Two-Dimensional Logarithmic Search) uses cross “+” pattern; [3]; CSA (Cross-Search Algorithm)[4] and DSWA(Dynamic Search-Window Adjustment) [5] adopt “X” and “+” pattern; TSS(Three-Step Search), NTSS (New TSS) [6], 4SS (Four-Step Search) [7], and BBGDS(Block-Based Gradient Descent Search)[8] employ square pattern; DS(Diamond Search)

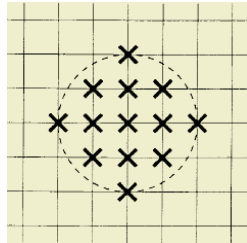
exploits diamond pattern [9]; HEXBS(Hexagon-Based Search) adopts hexagon pattern [10]; etc.

Based on the study of search patterns used in many fast BMAs, we proposed a fast BMA, which is based on diamond and triangle search patterns, in this paper.

The remainder of this paper is organized as follows. In section 2, we briefly analyze the DS algorithm. The proposed DTS algorithm is described in Section 3. Simulation results are presented in Section 4. Finally, a conclusion is given in the last section.

## 2 Analysis of DS Algorithm

The DS algorithm is one of the most famous fast BMAs, it was adopted and incorporated in MPEG-4 verification model. S. Zhu et al. pointed out in [9] that over 50% of the motion vectors are enclosed in a circular area with a radius of 2 pixels and centered on the position of zero motion as illustrated in Fig. 1.



**Fig. 1.** Motion vectors distribution

The DS algorithm employs two search patterns as illustrated in Fig.2. The first pattern, called large diamond search pattern (LDSP), comprises nine checking points from which eight points surround the center one to compose a diamond shape. The second pattern consisting of five checking points forms a smaller diamond shape, called small diamond search pattern (SDSP). In the searching procedure of the DS algorithm, LDSP is repeatedly used until the step in which the minimum block distortion (MBD) occurs at the center point. The search pattern is then switched from LDSP to SDSP as reaching to the final search stage. Among the five checking points in SDSP, the position yielding the MBD provides the motion vector of the best matching block.

By analyzing the DS algorithm, we found that it has the following limitations:

- It doesn't utilize the directional characteristic of SAD(Sum of Absolute Difference) distribution fully, so it will waste partial time to find the best matching block with large motion.
- It doesn't employ the center-biased characteristic of motion vectors(i.e. most of the motion vectors are centered on the position of zero motion), so it will take thirteen search points to find the best matching block with zero motion, and in this case, the ideal search points is only five.

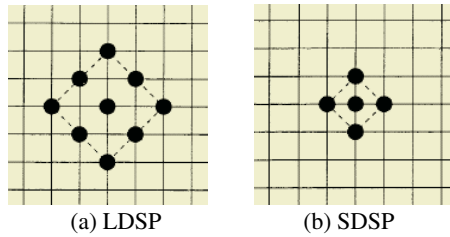


Fig. 2. Two search patterns employed in the DS algorithm

In short, the efficiency of the DS algorithm is not so high. So we developed a novel fast BMA, DTS, in this paper.

### 3 Diamond-Triangle Search Algorithm

#### 3.1 DTS Patterns

The proposed DTS algorithm adopts two search patterns adaptively in the process of motion search. The first pattern, called DP(Diamond Pattern, as shown in Fig.3(a)), comprises five checking points from which four points surround the center one to compose a diamond shape. The second pattern consisting of three checking points and covering the MBD point obtained in the previous search step(as shown in Fig.3(b)) forms a triangle shape, called TP(Triangle Pattern). In the process of motion search, DP is used to refine the motion vectors and it is necessary no matter than the motion vector being small or big, while TP is used to locate the best matching block with large motion approximately and it can be disused if the motion vector is zero.

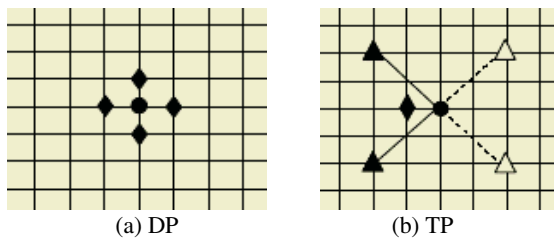


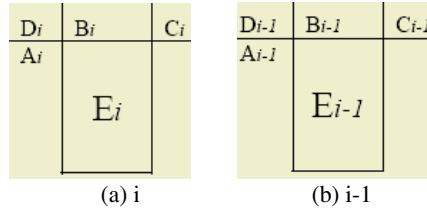
Fig. 3. Two search patterns employed in the proposed DTS algorithm

#### 3.2 DTS Algorithm

The proposed DTS algorithm has the following technical characteristics. Firstly, the initial search center is formed according to the predicted motion vector of the current block by the adjacent blocks. Secondly, DP and TP are adaptively employed according to the motion extents of macro blocks. The details about DTS algorithm are described as follows.

**1) Predict the initial search point by the adjacent blocks**

In order to reduce the search points for the best matching block with large motion, we use the median motion value of the adjacent blocks (as Fig.4 shows) to predict the motion vector of the current block. The median prediction is expressed as formula (1).



**Fig. 4.** Reference block location for predicting motion vector

$$pred\_mv = median(mv\_A, mv\_B, mv\_C) \tag{1}$$

- If  $A_i$  and  $D_i$  are outside the picture, their values are assumed to be zero.
- If  $D_i$ ,  $B_i$ , and  $C_i$  are outside the picture, the prediction is equal to  $A$ .
- If  $C_i$  is outside the picture or still not available due to the order of vector data,  $C_i$  is replaced by  $D_i$ .
- If  $A_i$ ,  $B_i$ ,  $C_i$  and  $D_i$  are outside the picture, the prediction is equal to the value of the co-located block in the previous frame (i.e.  $E_{i-1}$  as shown in Fig.4 (b)).

**2) Employ DP and TP adaptively according to motion extents of macro blocks**

Based on the assumption that most of the macro blocks in an image sequence of real world would be quasi-stationary or stationary, the DTS algorithm adopts DP at the first search step. If the best matching motion vector is zero, the DTS algorithm needs only 5 search points to find out. Otherwise, in order to judge whether the best matching motion vector is large or not, the DTS algorithm employs the TP which covers the MBD point obtained in the previous step at the second search step. If the new MBD point is located in one of the vertexes of DP, it indicates that the best matching motion vector would be small, DP is repeatedly used until the step in which the MBD occurs at the search center point. Otherwise, do the same as the beginning.

The DTS algorithm is summarized as follows.

**Preparation:** Use Formula (1) to predict the initial motion vector of the current block, and set the initial search center point according to the predicted value.

**Step1:** Dispose DP at the search center, and test the 5 checking points of DP (as shown in Fig. 3(a)). If the MBD point calculated is located at the center position, go to **step4**; otherwise, go to **step2**.

**Step2:** Dispose TP at the search center according to the MBD point obtained in the previous step, and test the 2 checking points of the TP which covers the MBD point obtained in the previous search step (as shown in Fig.3 (b)). If the new MBD point is



still located at one of the four vertexes of DP, go to **step3**; otherwise, the new MBD point is re-positioned as the search center point, go to **step1**.

**Step3:** The new MBD point found in the previous search step is re-positioned as the search center point. Dispose DP at the search center, and test the 3 checking points of DP. If the MBD point calculated is located at the center position, go to **step4**; otherwise, recursively repeat this step.

**Step4:** Stop searching. The center point is the final solution of the motion vector which points to the best matching block.

### 3.3 Analysis of the Proposed DTS Algorithm

For BMME, computational complexity could be measured by average number of search points required for each motion vector estimation. According to the statistical distribution law of motion vectors in different images sequences, assume that the best matching point is located in the circle area as shown in Fig.1, the least search points needed for DS and DTS are listed in Table 1.

**Table 1.** Comparison of search points near the initial search center for DS and DTS

	center	the best matching point is located in the		
		Circular area with a Radius of 1 pixel	Circular area with a Radius of $\sqrt{2}$ pixels	Circular area with a Radius of 2 pixels
DS	13	13	16	18
DTS	5	10	12	13

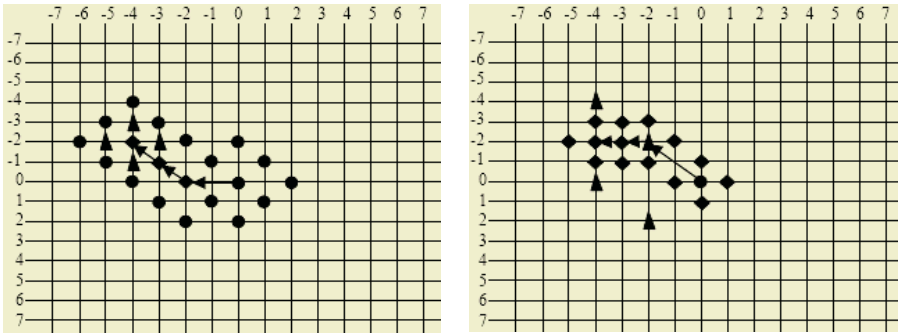
From Table 1 we observe that the least search points needed for DTS is always less than that of DS, and the reduced search points is always 3-8.

If the best matching point is located outside the circular area with a radius of 2 pixels, the least search points needed for DTS is still less than that of DS. This could be seen from the practical search path. Fig.5 gives a search path example which leads to the motion vector (-4,-2) for DS and DTS.

Although the average number of searched points can reflect the computational complexity of motion search, we use the CPU time (i.e. CPU clock cycles/frequency) consumed by the BMME to measure its computation complexity in practice for the fairness. In order to compare the speed of DTS with other BMAs, we use the speed improvement ratio (SIR) which is defined as follow:

$$SIR = \frac{T_{refBMA} - T_{DTS}}{T_{refBMA}} \times 100\% \tag{2}$$

Where  $T_{DTS}$  represents the total CPU time consumed by the proposed DTS algorithm, and  $T_{refBMA}$  denotes the total CPU time consumed by the reference block match motion estimation algorithm. The CPU time is measured by accumulating the CPU clock cycles occupied by the corresponding block-match motion estimation algorithm.



(a) DS uses five search steps – four times of LDSP and one time SDSP at the final step. There are 24 search points in total –taking nine, five, three, three, and four search points at each step, sequentially.

(b) DTS uses six search steps – two times of TP and four times of DP. There are 19 search points in total – taking five, two, four, two, three, and three search points at each step, sequentially.

**Fig. 5.** Search path example which leads to the motion vector (-4,-2) for DS and DTS

**Table 2.** SIR values of our DTS algorithm versus FS and DS

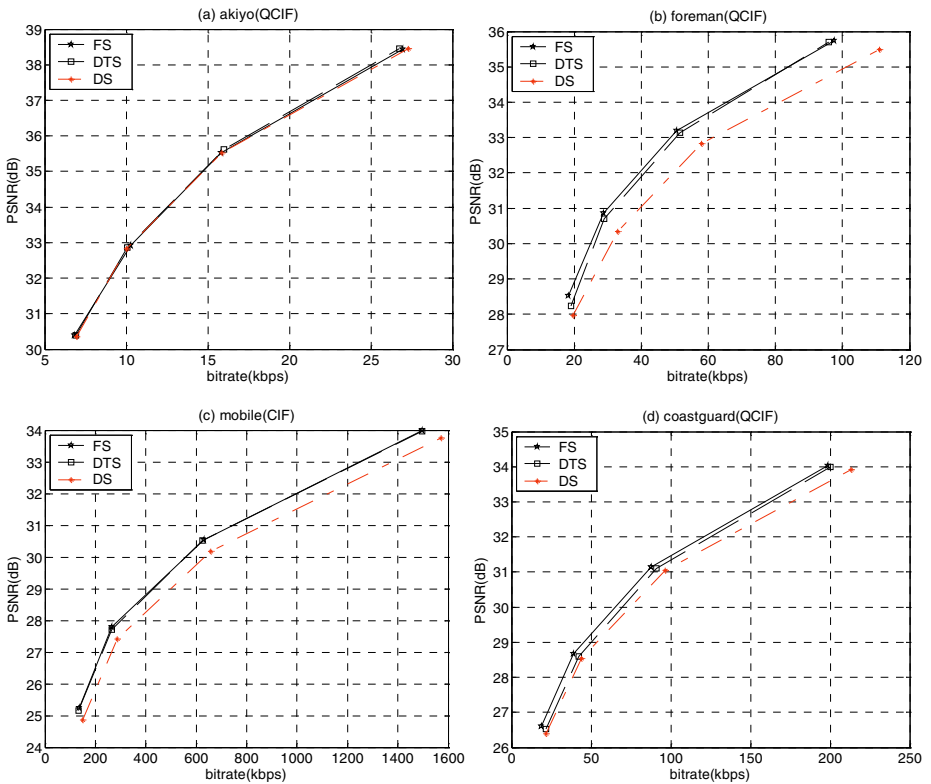
	DTS/FS				DTS/DS			
	QP=28	QP=32	QP=36	QP=40	QP=28	QP=32	QP=36	QP=40
Akiyo	98.25	98.09	97.85	97.66	42.80	44.17	44.66	46.96
Foreman	98.49	98.32	98.04	97.77	32.34	32.90	32.57	31.57
Mobile	99.03	98.89	98.69	98.50	31.97	31.09	30.75	26.71
Coastguard	99.00	98.86	98.71	98.60	33.35	33.72	32.24	35.22

### 4 Simulation Results

Our proposed DTS algorithm was integrated within version 7.6 of the H.264 software [11], and it is compared versus FS, and DS. Even though many image sequences are tested in the experiment, only four of them are selected out to be compared. The CABAC(Context-Adaptive Binary Arithmetic Coding) entropy coder [12] was used for all of our tests, with quantization parameter (QP) values of 28, 32, 36, and 40, a search range of  $\pm 32$ , and 2 references.

The four selected sequences are Akiyo(Quarter Common Intermediate Format, QCIF), Foreman(QCIF), Mobile(CIF), and Coastguard(QCIF). The former 100 frames of every sequence are tested, and only the first frame was encoded as I(inter)-frame, while the remainders are encoded as P(predictive)-or B(bi-predictive)- frames. The video sequence type is IBBPBBP.... To simplify our comparison, we have used SIR(Speed Improvement Ratio) and RD(Rate Distortion) performance plot as shown in Table 2 and Fig.6 respectively.

From Table 2 and Fig.6 we can observe that the computational complexity of the proposed DTS algorithm decreased about 97.66% to 99.03% with about 0.06dB on average loss in PSNR (Peak Signal to Noise Rate) compared with that of FS; or about



**Fig. 6.** RD performance plot for sequences (a) akiyo, (b) foreman, (c) mobile, and (d) coastguard

26.71% to 46.93% with about 0.17dB on average gain in PSNR compared with that of DS. Although the proposed DTS algorithm can also be trapped in local minima, the experimental results demonstrate that it is faster and better than DS.

### 5 Conclusions

Based on the directional characteristic of SAD distribution and the center-biased characteristic of motion vectors, a fast BMA, DTS, is proposed in this paper. The proposed DTS algorithm adaptively employs TP to locate the best matching block with large motion approximately, and DP to refine the motion vectors. Experimental results show that the proposed DTS algorithm can reduce the computational complexity of the BMME remarkably while incurring little, if any, loss in quality.

### Acknowledgements

This work was supported by Grant No.60173040 from the National Science Foundation of China and Grant No.04B055 from the Scientific Research Fund of Hunan Provincial Education Department.

## References

1. K. R. Rao and J. J Hwang.: Techniques and Standards for Image, Video and Audio Coding. Englewood Cliffs, NJ: Prentice Hall, 1996
2. T. Wiegand and G. Sullivan.: Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC), document JVT-G050d35.doc, 7th Meeting: Pattaya, Thailand, March 2003
3. J.Jain and A. Jain.: Displacement measurement and its application in interframe image coding. IEEE Transaction on Communication, vol. 29, pp.1799-1808, 1981
4. M. Ghanbari.: The cross-search algorithm for motion estimation. IEEE Transaction on Communication, vol. 38, pp. 950–953, July 1990
5. L. W. Lee, J. F. Wang, et al.: Dynamic search-window adjustment and interlaced search for block-matching algorithm. IEEE Transaction on Circuits and Systems for Video Technology, Vol. 3, pp.85–87, 1993
6. R.Li, B. Zeng, et al.: A new three-step search algorithm for block motion estimation. IEEE Transaction on Circuits and Systems for Video Technology, vol. 4, pp. 438–442, Aug. 1994
7. L. M. Po and W. C. Ma.: A novel four-step search algorithm for fast block motion estimation. IEEE Transaction on Circuits and Systems for Video Technology, vol. 6, pp. 313–317, June 1996
8. L. K. Liu and E. Feig.: A block-based gradient descent search algorithm for block motion estimation in video coding. IEEE Transaction on Circuits and Systems for Video Technology, vol. 6, pp. 419–423, Aug. 1996
9. S. Zhu and K. K. Ma.: A new diamond search algorithm for fast block matching motion estimation. IEEE Transaction on Image Processing, vol. 9, pp.287–290, Feb. 2000
10. C. Zhu, X. Lin, and L.P. Chau.: Hexagon-based search pattern for fast block motion estimation .IEEE Transaction on Circuits and Systems for Video Technology, vol.12, pp.349-355. May 2002
11. [http://bs.hhi.de/~suehring/tml/download/old\\_jm/jm7.6](http://bs.hhi.de/~suehring/tml/download/old_jm/jm7.6), June 2004
12. D. Marpe, H. Schwarz and T. Wiegand.: Context-Based Adaptive Binary Arithmetic Coding in H.264/AVC Video Compression Standard. IEEE Transaction on Circuits and Systems for Video Technology, vol.13, pp. 620–636, July 2003

# A Watermarking Scheme Based on Discrete Non-separable Wavelet Transform

Jianwei Yang<sup>1,2</sup>, Xinge You<sup>1,3</sup>, Yuan Yan Tang<sup>1</sup>, and Bin Fang<sup>1</sup>

<sup>1</sup> Department of Computer Science

Hong Kong Baptist University Kowloon Tong, Hong Kong

{yjianw,xyou,yytang,fangb}@comp.hkbu.edu.hk

<sup>2</sup> Department of Computer Science, Henan Institute of Finance and Economics  
Zhengzhou, P.R. China

<sup>3</sup> Faculty of Mathematics and Computer Science, Hubei University  
Wuhan, P.R. China

**Abstract.** This paper presents a novel method for constructing non-separable wavelet filters. The high frequency sub-bands of non-separable wavelet transform can reveal more features than that of the common used separable wavelet transform. Then, we describe a blind watermarking scheme which is based on discrete non-separable wavelet transform (DNWT). More coefficients of DNWT can add watermark than that of discrete separable wavelet transform (DSWT). Experiment results show that the DNWT watermarking scheme is robust to noising, JPEG compression, and cropping. Especially, it is more resistant to sharpening than DSWT scheme. Furthermore, by adjusting the threshold such that the number of the DSWT coefficients to embed watermark is not less than the number of the DNWT coefficients, the performance of DSWT to sharpening is still worse than the DNWT. Such adjustment also dramatically decreases the robustness of the DSWT scheme to noising.

## 1 Introduction

Digital watermarking has been proposed to solve the problem of copyright protection. Recent research has shown that watermarking in wavelet domain has some advantages over other watermarking approaches (see [2], [4], [5] and references therein). Almost all the literatures of wavelet based watermarking schemes use discrete separable wavelet transform (DSWT) to embed watermark. However, the property of anisotropy makes separable wavelet unattractive for the purpose of watermarking, which demands the extraction of more features of the image. The high frequency sub-bands of non-separable wavelet transform can reveal more features than that of the common used separable wavelet transform. We propose to embed watermark by discrete non-separable wavelet transform (DNWT) in this paper.

Many efforts have been spent on constructing non-separable wavelets (see [3] and references therein). However, up to now, there is still no systematic method to construct two-dimensional non-separable wavelets. Even for the construction

of non-separable wavelets supported over  $[0, 3] \times [0, 3]$ , we have to deal with an implicit constrained condition (see Eq. (11) in [3]).

In this paper, we present a novel method for constructing two dimensional orthogonal wavelet filters begin with one dimensional wavelet filters but non-separable can be achieved.

The non-separable wavelet filters derived from our method are applied for watermarking still images. We describe a blind watermarking scheme which is based on the method given by Dugad [2] but embeds the watermark by DNWT. We add the watermark to all coefficients in the high frequency sub-bands above a threshold. For the same threshold, watermark (pseudo-random codes) will be added to more coefficients in the DNWT watermarking scheme than that in the DSWT one. Experiments show that the DNWT watermarking scheme is robust to some distortions such as noising, JPEG compression, and cropping. In particular, it is more resistant to sharpening than DSWT scheme. By adjusting the threshold such that the number of the DSWT coefficients to embed watermark is not less than the number of the DNWT coefficients, the response on sharpening is still worse than the DNWT. However, such adjustment will dramatically decrease the robustness of the DSWT scheme to noising.

We present a method for constructing non-separable wavelet filters by one dimensional orthogonal wavelet filters in Section 2. In Section 3, we describe the watermark embedding approach and the detection method. Experimental results are presented in Section 4. The conclusion is given in Section 5.

## 2 Construction of Non-separable Wavelet Filters

To construct two-dimensional orthogonal scaling function  $\phi(x, y)$ , we need to construct *orthogonal low-pass wavelet filter*  $\{p_{(k_1, k_2)}\}$  such that

$$\sum_{k_1, k_2 \in Z} p_{(k_1, k_2)} = 4, \quad \text{and} \quad \sum_{k_1, k_2} p_{(k_1, k_2)} p_{(k_1+2\gamma_1, k_2+2\gamma_2)} = 4\delta_{0, \gamma_1} \delta_{0, \gamma_2}. \quad (1)$$

To construct the associated orthogonal wavelets  $\psi^j(x, y)$  ( $j = 1, 2, 3$ ), we need to construct *orthogonal high-pass wavelet filters*  $\{q_{(k_1, k_2)}^j\}$  ( $j = 1, 2, 3$ ) such that

$$\sum_{k_1, k_2} p_{(k_1, k_2)} q_{(k_1+2\gamma_1, k_2+2\gamma_2)}^l = 0, \quad (2)$$

$$\sum_{k_1, k_2} q_{(k_1, k_2)}^{l_1} q_{(k_1+2\gamma_1, k_2+2\gamma_2)}^{l_2} = 4\delta_{0, \gamma_1} \delta_{0, \gamma_2} \delta_{l_1, l_2}, \quad (3)$$

where  $l, l_1, l_2 = 1, 2, 3$ , and  $\gamma_1, \gamma_2 \in Z$ . A sequence  $\{p_{(k_1, k_2)}\}$  is called *non-separable low-pass wavelet filter* if it satisfies (1), and its mask  $P(\omega_1, \omega_2) = \frac{1}{4} \sum_{k_1, k_2} p_{(k_1, k_2)} e^{-i\omega_1 k_1} e^{-i\omega_2 k_2}$  is nonseparable.

Let  $\varphi(x)$  be the compactly supported one-dimensional orthogonal scaling function, which satisfies the equation

$$\varphi(x) = \sum_{n \in Z} h_n \varphi(2x - n) \quad (4)$$

where  $\{h_n\}$  is a finite sequence, such that  $\sum_{n \in \mathbb{Z}} h_n h_{n+2s} = 2\delta_{0,s}$ . Then  $\psi(x) = \sum_{n \in \mathbb{Z}} g_n \varphi(2x - n)$  is the associated orthogonal wavelet, where  $g_n = (-1)^n h_{1-n}$ .

**Theorem 1.** For  $k = 1, 2$ , choose integers  $a_k, b_k, c_k, d_k, s_k, t_k, l_k$  and  $u_k$ , such that  $|a_k| = |b_k| = |c_k| = d_k$ , plus  $s_k + t_k$  and  $l_k + u_k$  are odd. Define the sequence  $\{p_{(k_1, k_2)}\}$  as follows:

$$p_{(i,j)} = \begin{cases} h_{2n}h_{2m} & \text{if } i = 2a_1n + s_1 \text{ and } j = 2a_2m + s_2, \\ h_{2n+1}h_{2m} & \text{if } i = 2b_1n + t_1 \text{ and } j = 2c_2m + l_2, \\ h_{2n}h_{2m+1} & \text{if } i = 2c_1n + l_1 \text{ and } j = 2b_2m + t_2, \\ h_{2n+1}h_{2m+1} & \text{if } i = 2d_1n + u_1 \text{ and } j = 2d_2m + u_2, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

If the difference of any two evens and any two odd in  $s_k, l_k, t_k, u_k$  can be divided by  $a_i$ , then  $\{p_{(k_1, k_2)}\}$  defined above is an orthogonal low-pass wavelet filter.

**Theorem 2.** Suppose that  $\{p_{(k_1, k_2)}\}$  is a sequence constructed in Theorem 1. We define the sequences  $\{q_{(k_1, k_2)}^j\}$  ( $j = 1, 2, 3$ ) as follows

$$q_{(i,j)}^1 = \begin{cases} h_{2n}g_{2m} & \text{if } i = 2a_1n + s_1 \text{ and } j = 2a_2m + s_2, \\ h_{2n+1}g_{2m} & \text{if } i = 2b_1n + t_1 \text{ and } j = 2c_2m + l_2, \\ h_{2n}g_{2m+1} & \text{if } i = 2c_1n + l_1 \text{ and } j = 2b_2m + t_2, \\ h_{2n+1}g_{2m+1} & \text{if } i = 2d_1n + u_1 \text{ and } j = 2d_2m + u_2, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

$$q_{(i,j)}^2 = \begin{cases} g_{2n}h_{2m} & \text{if } i = 2a_1n + s_1 \text{ and } j = 2a_2m + s_2, \\ g_{2n+1}h_{2m} & \text{if } i = 2b_1n + t_1 \text{ and } j = 2c_2m + l_2, \\ g_{2n}h_{2m+1} & \text{if } i = 2c_1n + l_1 \text{ and } j = 2b_2m + t_2, \\ g_{2n+1}h_{2m+1} & \text{if } i = 2d_1n + u_1 \text{ and } j = 2d_2m + u_2, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

$$q_{(i,j)}^3 = \begin{cases} g_{2n}g_{2m} & \text{if } i = 2a_1n + s_1 \text{ and } j = 2a_2m + s_2, \\ g_{2n+1}g_{2m} & \text{if } i = 2b_1n + t_1 \text{ and } j = 2c_2m + l_2, \\ g_{2n}g_{2m+1} & \text{if } i = 2c_1n + l_1 \text{ and } j = 2b_2m + t_2, \\ g_{2n+1}g_{2m+1} & \text{if } i = 2d_1n + u_1 \text{ and } j = 2d_2m + u_2, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Then,  $\{q_{(k_1, k_2)}^j\}$  ( $j = 1, 2, 3$ ) satisfying Eq. (2) and (3), are the high-pass wavelet filters associated with  $\{p_{(k_1, k_2)}\}$ .

When compared with [3], our method given in (5)~(8) does not need to solve an implicit constrained condition (see Eq. (11) in [3]), and the the highpass & lowpass wavelet filters are given in explicit expression (see (5)~(8)).

**Theorem 3.** Let  $\varphi(x)$  be a one-dimensional orthogonal scaling function satisfying Eq. (4).  $\{h_n\}$  is a sequence associated with the one-dimension scaling

function  $\varphi(x)$  as in (4). Suppose that there exists  $k, s \in Z (s \geq 1)$ , such that  $h_k h_{k+2s+1} \neq h_{k+1} h_{k+2s}$ . Choose integers  $a_k, b_k, c_k, d_k, l_k, u_k, s_k, t_k$ , such that

$$a_k = b_k = c_k = d_k = 1, \quad s_2 = l_k = t_2 = u_k = 0, \quad k = 1, 2 \quad (9)$$

$$s_1 = 1, \quad \text{and} \quad t_1 = -1. \quad (10)$$

Then, the filter  $\{p_{(k_1, k_2)}\}$ , which is given by (5), is a orthogonal non-separable low-pass wavelet filter.

*Remark 1.* Theorem 3 provides a concrete algorithm for constructing non-separable orthogonal wavelet filters based on one-dimensional wavelet filters. Here, we only give the construction under the conditions (9) and (10). In many other cases, non-separable wavelet filters can also be derived.

### 3 Watermarking by DNWT

The watermark is embedded into the image according to the following procedure:

**Step 1** Compute the DNWT coefficients of the original image. In this paper, we compute DNWT coefficients at three levels.

**Step 2** Add watermark to those coefficients whose magnitudes are greater than a given threshold ( $T_1$ ) in the sub-bands other than the low pass sub-band. The equations used for watermark casting and detection are given as follows:

$$\hat{S}_i = S_i + \alpha |S_i| x_i, \quad (11)$$

where  $i$  runs over all DNWT coefficients whose magnitudes are greater than  $T_1$  (except the low-pass component).  $S_i$  denotes the corresponding DNWT coefficients of the original image and  $\hat{S}_i$  denotes the DNWT coefficients of the watermarked image.  $x_i$  are the watermark values for each component of  $S_i$ , which are generated from a normal distribution of zero mean and unit variance.  $\alpha$  is a parameter to control the intensity of the watermark. In this paper we use an appropriate value of scale factor  $\alpha$  for each sub-band.

**Step 3** Compute the inverse DNWT to reconstruct the watermarked image.

The watermark detector is correlation-based, similar to Dugad [2]. All the high pass coefficients whose magnitudes greater than  $T_2$  are chosen, and are correlated with the original copy of the watermark.  $T_2$  is larger than  $T_1$ . The correlation  $z$  between the DNWT coefficients  $\hat{S}_i$  of the corrupted watermarked image and a test watermark is computed as  $z = \frac{1}{M} \sum_i \hat{S}_i y_i$ , where  $y_i$  stands for the value of the  $i$ -th watermark component,  $i$  is an index over all the significant coefficients of the input image and  $M$  is the total number of such coefficients. The threshold  $G$  is defined as

$$G = \frac{1}{2M} \left( \sum_t \sum_i \alpha_t |\hat{S}_i^t| \right), \quad (12)$$

where  $\alpha_t$  denotes the scale factor for level  $t$ , and  $\hat{S}_i^t$  denote the DNWT coefficients of the corrupted watermarked image of level  $t$ . If  $z \geq G$ , the watermark is present in the input image.



## 4 Experimental Results

First, we provide the non-separable wavelet filters adopted in our experiments. Consider the scaling function  $\varphi(x)$  which satisfies the equation [1]:  $\varphi(x) = \sum_{i=1}^3 h_i \varphi(2x - i)$ , where  $h_0 = \frac{1+\sqrt{3}}{4}$ ,  $h_1 = \frac{3+\sqrt{3}}{4}$ ,  $h_2 = \frac{3-\sqrt{3}}{4}$ , and  $h_3 = \frac{1-\sqrt{3}}{4}$ . By Theorem 3, we construct the non-separable low-pass wavelet filter  $\{p_{(k_1, k_2)}\}$ :

$$\begin{array}{llll}
p_{(0,0)} = \frac{6+4\sqrt{3}}{32} & p_{(0,1)} = \frac{12+6\sqrt{3}}{32} & p_{(0,2)} = \frac{6}{32} & p_{(0,3)} = -\frac{2\sqrt{3}}{32} \\
p_{(1,0)} = \frac{4+2\sqrt{3}}{32} & p_{(1,1)} = \frac{6+4\sqrt{3}}{32} & p_{(1,2)} = \frac{2\sqrt{3}}{32} & p_{(1,3)} = -\frac{2}{32} \\
p_{(2,0)} = -\frac{2}{32} & p_{(2,1)} = -\frac{2\sqrt{3}}{32} & p_{(2,2)} = \frac{6-4\sqrt{3}}{32} & p_{(2,3)} = \frac{4-2\sqrt{3}}{32} \\
p_{(3,0)} = \frac{2\sqrt{3}}{32} & p_{(3,1)} = \frac{6}{32} & p_{(3,2)} = \frac{12-6\sqrt{3}}{32} & p_{(3,3)} = \frac{6-4\sqrt{3}}{32}
\end{array}$$

The associated high-pass filters can be given by Theorem 2 in the similar way.

**Table 1.** Comparison of DSWT and DNWT: the number of coefficients whose magnitudes greater than a threshold ( $T_1 = 40$ ) in the high frequency sub-bands by DNWT and DSWT respectively.

	lena	boat	mandrill	peppers	goldhill
DSWT	6110	8561	23946	5884	6071
DNWT	9103	12689	27147	8972	8812

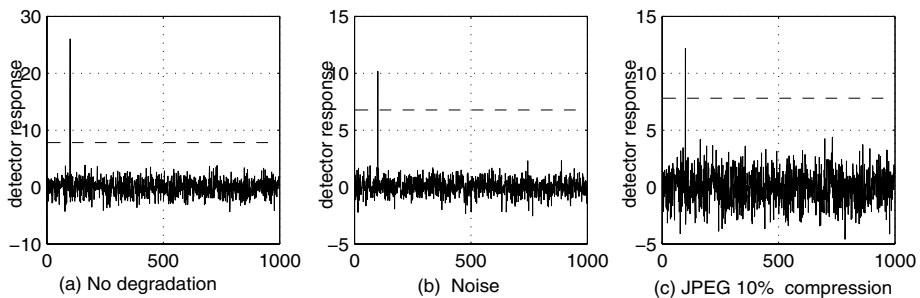
The high frequency sub-bands of non-separable wavelet transform can reveal more features than that of the common used separable wavelet transform. Table 1 shows the numbers of coefficient whose magnitudes greater than a threshold ( $T_1 = 40$ ) in the high frequency sub-bands by DNWT and that by DSWT respectively. Each image is gray scale and has size of  $512 \times 512$ . It was observed that effective numbers of coefficient in DNWT are larger than that in DSWT. Hence, more wavelet coefficients will be involved in our DNWT watermarking scheme than in the DSWT one.

Our DNWT watermarking scheme is based on the above filters. The threshold for significance of wavelet coefficients for the watermark embedder is  $T_1 = 40$  and the detector is  $T_2 = 50$ . Different values of scale factor  $\alpha$  ( $\alpha = 0.14, 0.18, 0.20$ ) are used for level 3, 2, and 1 respectively. The test image is the gray scale *lena* image of size  $512 \times 512$ . Figure 1(a) displays the original *lena* image and Figure 1(b) illustrates the watermarked *lena* image by using the above non-separable wavelet filters  $\{p_{(k_1, k_2)}\}$ , and the associated highpass wavelet filters  $\{q_{(k_1, k_2)}^l\}$  ( $l = 1, 2, 3$ ). The distortion of the watermarked image is not recognizable by visual inspection, which validates the effectiveness of the proposed DNWT watermarking.

We tested our DNWT scheme's robustness against various attacks. In the experiments, 1000 watermarks (i.e., pseudo-random sequences) have been generated randomly, only one of them is the watermark embedded in image. Figure 2(a) shows the response of the watermark detector to 1000 randomly generated



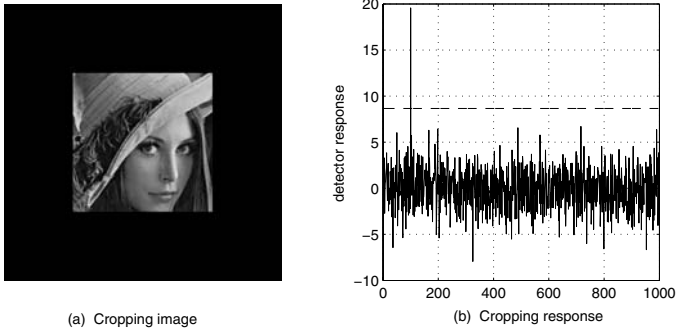
**Fig. 1.** (a) The original “lena” image; (b) Watermarked “lena” image by DNWT.



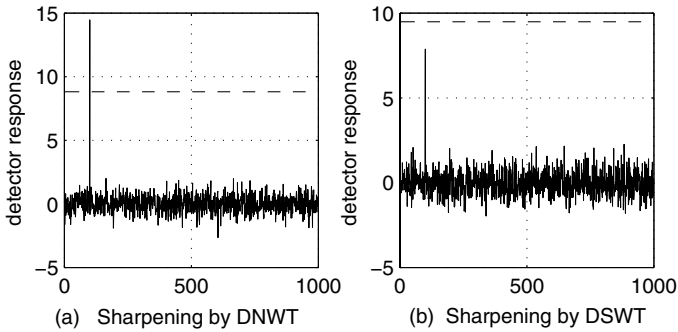
**Fig. 2.** (a) Responses on the watermarked image (Fig. 1(b)); (b) Responses to Gaussian noise with  $\sigma^2 = 500$ ; (c) Responses to JPEG 10% quality compression.

watermarks. We find that the response of the correct watermark (for example 100 in the figure) is much stronger than the response to incorrect watermarks. Fig.2 (b), Fig.2 (c), Fig. 3 (b), Fig.4 (a) show the responses of our DNWT scheme’s against noising, JPEG compression, cropping and sharpening respectively. Results show that we can still correctly detect the watermark under these attacks.

It is observed that Dugad’s technique is unable to embed the watermark into many coefficients of the image [5], and their method is vulnerable to attacks such as sharpening (see Fig. 4(b)). When compared with the method given in [2], our DNWT watermarking scheme is much more resistant to the attack of sharpening than the scheme given by [2] (see Fig. 4(a)). To improve the robustness of watermarking scheme in [2] against sharpening, Oriol Guitart Pla et.al.[5] describe a watermarking scheme by using the tree structure of the DSWT. The method is more complex and computationally expensive. In comparison with [5], our



**Fig. 3.** (a) A Cropped image, which retains only the central portion. (b) Responses to the cropped image by DNWT.



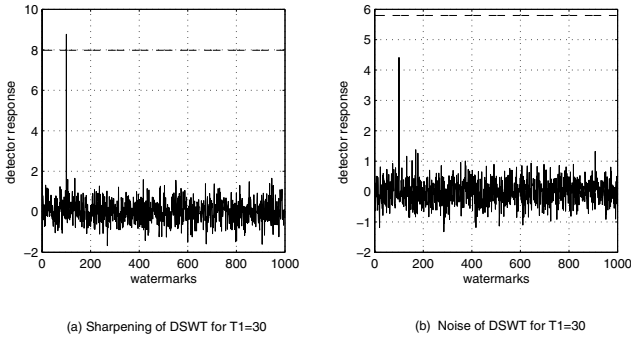
**Fig. 4.** Responses to sharpening attack: (a) by DNWT; (b) by DSWT.

method is much efficient in computation since we do not need to compute the children of the significant coefficients.

We also perform some experiments to see if the DNWT watermarking scheme can be substituted by DSWT watermarking scheme. We decrease the threshold  $T_1$  (40 to 30) in DSWT scheme, then there are 9123 DSWT coefficients being embedded with watermark, which is bigger than the number of coefficients (9103) in our DNWT method for threshold 40. In Fig. 5(a), it is shown that although the performance of DSWT to sharpening is improved, it is still worse than DNWT (see Fig. 4(a)). Furthermore, such adjustment will dramatically decrease the robustness of the DSWT watermarking scheme to noising (see Fig. 5(b)).

## 5 Conclusion

A novel method is presented for constructing non-separable wavelet filters. Based on the fact that the high frequency sub-bands of non-separable wavelet transform can reveal more features than that of the separable one, we propose to embed watermark by DNWT and describe a blind watermarking scheme. Experiments



**Fig. 5.** (a) Responses to the sharpening attack of DSWT for  $T_1 = 30$ : (b) Response to the nosing attack of DSWT for  $T_1 = 30$ .

show that this scheme is robust to noising, JPEG compression, and cropping, especially for sharpening.

## Acknowledgments

This research was partially supported by a grant (60403011 and 10371033) from National Natural Science Foundation of China and a grant (2003ABA012) from Hubei Provincial Science & Technology Department, and a grant (20045006071-17) from Wuhan government, China. This research was also supported by the grants (RGC and FRG) from Hong Kong Baptist University.

## References

1. Daubechies, I.: Ten lectures on wavelets. CBMS-NSF Reg. Conf. Series on Appl. Math., Vol. 61. SIAM. Philadelphia (1992)
2. Dugad, R., Ratakonda, K., Ahuja, N.: A new wavelet-based scheme for watermarking images. In: Proceedings of the IEEE International Conference on Image Processing (Chicago), Vol. II. (1998)
3. He, W., Lai, M.: Examples of bivariate nonseparable compactly supported continuous wavelets. *IEEE Trans. Image Process.* **9** (2000) 949–953
4. Lu, C.-S., Liao, H.-Y.M.: Multipurpose watermarking for image authentication and protection. *IEEE Trans. on image Process.* **10** (2001) 1579–1592
5. Oriol Guitart Pla, Delp, E. J.: A wavelet watermarking algorithm based on a tree structure. In: Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, Vol. VI. (2004) 19 – 22

# A Fast Run-Length Algorithm for Wavelet Image Coding with Reduced Memory Usage

Jose Oliver and Manuel P. Malumbres

Department of Computer Engineering (DISCA)  
Technical University of Valencia  
Camino de Vera 17, 46017, Spain  
{jooliver,mperez}@disca.upv.es

**Abstract.** A new image coder is described in this paper. Since it is based on the Discrete Wavelet Transform (DWT), it yields good Rate/Distortion (R/D) performance. However, our proposal focuses on overcoming the two main problems of wavelet-based image coders: they are typically implemented by memory-intensive and time-consuming algorithms. In order to avoid these common drawbacks, we ought to tackle these problems in the main stages of this type of coder, i.e., both the wavelet computation and the entropy coding of the coefficients. The proposed algorithms are described in such a manner that they can be implemented in any programming language straightforwardly. The numerical results show that while the R/D performance achieved by our proposal is similar to the state-of-the-art coders, such as SPIHT and JPEG2000/Jasper, the amount of memory required in our algorithm is reduced drastically (in the order of 25 to 35 times less memory), and its execution time is lower (three times lower than SPIHT, and more than ten times lower than JPEG 2000/Jasper).

## 1 Introduction

Wavelet-based image coders have aroused great interest in the last years due to their nice features, such as natural multiresolution and high compactness of the coefficients, which leads to high compression efficiency. However, one of the main drawbacks of current wavelet encoders is their high memory usage, since the regular wavelet transform requires a lot of memory to be computed. In addition, in many wavelet encoders, the subsequent coding process uses some extra lists and introduces memory overhead. The complexity of these algorithms is another usual problem. In this paper, we deal with both problems (memory requirement and complexity) in both stages (wavelet transform and efficient coding).

## 2 Wavelet Transform for Image Coding with Low Use of Memory

One of the desirable features of the proposed image coder is to have low memory consumption. Since our proposal is a wavelet-based coder, the first bottleneck that appears in the efficient use of memory is the computation of the DWT. Our encoder

```

function GetLLlineBwd( level )

1) First base case: No more lines to be read at this level
   if LinesReadlevel = MaxLineslevel return EOL

2) Second base case: The current level belongs to the space domain and
   not to the wavelet domain
   else if level = 0 return ReadImageLineIO( )
   else

3) Recursive case
3.1) Recursively fill or update the buffer for this level
   if bufferlevel is empty
       for i = N ... 2N
           bufferlevel(i) = 1D_DWT(GetLLlineBwd( level-1))
           FullSymmetricExtension( bufferlevel )
       else
           repeat twice
               Shift( bufferlevel )
               line = GetLLlineBwd( level-1 )
               if line = EOL bufferlevel(2N) = SymmetricExt( bufferlevel )
               else bufferlevel(2N) = 1D_DWT( line )

3.2) Calculate the WT from the lines in the buffer, then process the result-
   ing subband lines (LL, HL, LH and HH)
   {LLline, HLline} = ColumnDWT_LowPass( bufferlevel )
   {LHline, HHline} = ColumnDWT_HighPass( bufferlevel )
   EncodeSubLines( {HLline, LHline, HHline}, level )
   set LinesReadlevel = LinesReadlevel + 1
   return LLline
end of fuction

```

**Algorithm 1.1.** Backward recursive function

could only have low memory consumption if the DWT is performed in an efficient way. In the regular DWT, Mallat decomposition is performed [1]. In this decomposition, the image is transformed first row by row, and then column by column, at every decomposition level. Therefore, it must be kept entirely in memory. In this section we propose a different wavelet transform in which the key idea for saving memory is to get rid of the wavelet coefficients as soon as they have been calculated.

This idea was first used in [2], aiming to reduce the memory requirements of the 1D DWT. In [3], this transform is extended to image wavelet transform (2D), and other issues related to the order of the data are solved. However, in this 2D version, the authors do not propose a direct algorithm to implement their proposal, and it cannot be easily implemented due to some unclear aspects. In [4], we presented a general-purpose recursive algorithm that we will use in the image coder presented in this

```

program Code_Image (nlevel, Q, rplanes)

    set LinesReadlevel = run_lengthlevel = 0  ∀level ∈ nlevel

    set MaxLineslevel =  $\frac{height}{2^{level}}$   ∀level ∈ nlevel

    set bufferlevel = EncBufferHLlevel = EncBufferLHlevel =
        EncBufferHHlevel = empty  ∀level ∈ nlevel

    repeat  $\frac{height}{2^{nlevel}}$  times
        LLline = GetLLlineBwd(nlevel)
        EncodeLLSubLine(LLline)

    end of program
    
```

**Algorithm 1.2.** Perform the DWT and encode the image by calling a backward recursive function (see Algorithm 1.1)

paper. In this section, this wavelet transform is outlined, while the reader is referred to [4] for a more complete and exhaustive description.

The proposed algorithm relies on a line-based strategy. In this strategy, we only keep in memory those image lines that we are dealing with. This way, there is a buffer in each level that is able to keep  $2N+1$  lines for the low-frequency subband (LL) at that level ( $2N+1$  is the length of the filter bank). These buffers are filled so that, when they are full, one-step of a column wavelet transform is performed. This operation generates a line of every wavelet subband (HH, HL and LH at that level), and a LL line. The HH, HL and LH lines can be directly encoded, while the LL line is passed to the following level in order to fill its buffer up.

The drawback of this algorithm is the synchronization among the buffers. Before a buffer can produce lines, it must be filled with lines from previous buffers, therefore they start working at different moments, i.e., they have different delays. Moreover, all the buffers exchange their result at different intervals, according to their level.

To solve the synchronization problem, we define a recursive function called GetLLlineBwd (*level*), which obtains the next LL line from a contiguous level. This algorithm is formally described in the frame *Algorithm 1.1*, while *Algorithm 1.2* defines the main program that sets up some variables and performs the image transform by calling the recursive function. Let us see the first algorithm more carefully.

The first time that the recursive function is called at every level, its buffer (*buffer<sub>level</sub>*) is empty and it has to be filled up. So, its upper half (from  $N$  to  $2N$ ) is recursively filled with lines from the previous level. When a line is received, it must be transformed using a 1D DWT before it is stored. The lower half part is filled using symmetric extension (the  $N+1$  line is copied into the  $N-1$  position ...)

On the other hand, if the buffer is not empty, it simply has to be updated. In order to update it, it is shifted one position so that a new line can be introduced in the last position ( $2N$ ) using a recursive call. This operation is repeated twice.

```

function EncodeSubLines( {HLline, LHline, HHline}, level )
  AddToBuffer ( EncBufferHLlevel, HLline )
  AddToBuffer ( EncBufferLHlevel, LHline )
  AddToBuffer ( EncBufferHHlevel, HHline )
  if IsFull ( EncBufferHLlevel )
    RLW_Code_Subband ( EncBufferHLlevel, level )
    RLW_Code_Subband ( EncBufferLHlevel, level )
    RLW_Code_Subband ( EncBufferHHlevel, level )
    EncBufferHLlevel = EncBufferLHlevel = EncBufferHHlevel = empty
  end of function

```

**Algorithm 2.1.** Store the subband lines in the encoder buffer and call the run-length coding function when they are full

However, if there are no more lines in the previous level, this recursive call will return *End Of Line* (EOL). That points out that we are about to finish the computation at this level, but we still need to fill the buffer up using symmetric extension again.

Once the buffer is filled or updated, both high-pass and low-pass filter banks are applied to every column in the buffer. This way, we get a line of every wavelet subband at this level, and a LL line. The wavelet coefficients are passed to the coder so that they can be compressed, and the function returns the LL line.

Notice that this function has two base cases. In the first one, all the lines at this level have been read. It is detected by keeping an account of the number of lines read, and it returns EOL. In the second one, the variable *level* reaches 0 and then no further recursive call is needed since an image line can be read directly. Moreover, the maximum recursion depth is given by the decomposition level (which is usually 5 or 6), and so the memory usage for recursion is negligible compared with the buffer sizes.

### 3 Run-Length Coding of the Wavelet Coefficients

In order to have low memory consumption, once a wavelet subband line is calculated, it has to be encoded as soon as possible to release memory. However, we cannot encode independent lines if we want good R/D performance, since entropy coders need to exploit local similarities in the image to be efficient. *Algorithm 2.1* stores the subband lines in encoder buffers so that when they are full, there are enough lines to perform an efficient compression, and the coding function is called.

The encoder cannot use global image information since it does not know the whole image. Moreover, we aim at fast execution, and hence no R/D optimization or bit-plane processing can be made, because it would turn it slower. In the next subsection, a Run-Length Wavelet (RLW) encoder with the aforementioned features is proposed.

#### 3.1 Fast Run-Length Coding

In the proposed algorithm, the quantization process is performed by two strategies: one coarser and another finer. The finer one consists on applying a scalar uniform



quantization to the coefficients using the  $Q$  parameter (see *Algorithm 1.2*). The coarser one is based on removing bit planes from the least significant part of the coefficients. We define  $rplanes$  as the number of less significant bits to be removed, and we call significant coefficient to those coefficients  $c_{i,j}$  that are different to zero after discarding the least significant  $rplanes$  bits, in other words, if  $c_{i,j} \geq 2^{rplanes}$ .

The wavelet coefficients are encoded as follows. The coefficients in the buffer are scanned column by column (to exploit their locality). For each coefficient in that buffer, if it is not significant, a run-length count of insignificant symbols at this level is increased ( $run\_length_L$ ). However, if it is significant, we encode both the count of insignificant symbols and the significant coefficient, and  $run\_length_L$  is reset.

The significant coefficient is encoded by means of a symbol indicating the number of bits required to represent that coefficient. An arithmetic encoder with two contexts is used to efficiently store that symbol. As coefficients in the same subband have similar magnitude, an adaptive arithmetic encoder is able to represent this information in a very efficient way. However, we still need to encode its significant bits and sign. They are raw encoded to speed up the execution time.

```

function RLW_Code_Subband( Buffer, L )
  Scan Buffer in horizontal raster order (i.e., in columns)
  for each  $c_{i,j}$  in Buffer
     $nbits_{i,j} = \lceil \log_2(|c_{i,j}|) \rceil$ 
    if  $nbits_{i,j} \leq rplanes$ 
      increase  $run\_length_L$ 
    else
      if  $run\_length_L \neq 0$ 
        if  $run\_length_L < enter\_run\_mode$ 
          repeat  $run\_length_L$  times
            arithmetic_output LOWER
          else
            arithmetic_output RUN
             $rbits = \lceil \log_2(run\_length_L) \rceil$ 
            arithmetic_output  $rbits$ 
            output  $bit_{rbits-1}(run\_length_L) \dots bit_1(run\_length_L)$ 
             $run\_length_L = 0$ 
            arithmetic_output  $nbits_{i,j}$ 
            output  $bit_{nbits_{i,j}-1}(|c_{i,j}|) \dots bit_{rplane+1}(|c_{i,j}|)$ 
            output  $sign(c_{i,j})$ 
  end of function
  Note:  $bit_n(c)$  is a function that returns the  $n^{th}$  bit of  $c$ 

```

**Algorithm 2.2.** Run-length coding of the wavelet coefficients

In order to encode the count of insignificant symbols, we encode a *RUN* symbol. After encoding this symbol, the run-length count is stored in a similar way as in the significant coefficients. First, the number of bits needed to encode the run value is arithmetically encoded (with a different context), afterwards the bits are raw encoded.

Instead of using run-length symbols, we could have used a single symbol to encode every insignificant coefficient. However, we would need to encode a larger amount of symbols, and therefore the complexity of the algorithm would increase (most of all in the case of large number of insignificant contiguous symbols, which usually occurs in moderate to high compression ratios).

Despite of the use of run-length coding, the compression performance is increased if a specific symbol is used for every insignificant coefficients, since an arithmetic encoder stores more efficiently many likely symbols than a lower amount of less likely symbols. So, for short-run lengths, we encode a *LOWER* symbol for each insignificant coefficient instead of coding a run-length symbol for all the sequence. The threshold to enter the run-length mode and start using run-length symbols is defined by the *enter\_run\_mode* parameter. The formal description of the depicted algorithm can be found in the frame entitled *Algorithm 2.2*.

### 3.2 Tradeoff Between R/D Performance and Speed and Memory Requirements

The proposed algorithm can be tuned according to the final application. Thus, some parameters can be adjusted to improve the compression performance at the cost of slightly higher memory requirements or execution time. This way, the size of the encoder buffer can be 8 subband lines for a good R/D performance, but compression efficiency can be improved with 16 lines, increasing the memory requirements. Another parameter that can be tuned is the *enter\_run\_mode* variable in *Algorithm 2.2*. When this parameter is increased, larger run-lengths are encoded by successive *LOWER* symbols, which results slower but a bit more efficient in R/D performance. Another tradeoff between compression and complexity is the use of an arithmetic encoder (with nine contexts) for the sign of the coefficients. In general, each of these improvements may increase the PSNR of an image encoded at 1bpp in about 0.1 dB, while the two latter improvements increase the execution time in about 20% each one.

## 4 Numerical Results

We have implemented the proposed coder in ANSI C language. In this section we will compare it with the state-of-the-art wavelets coders SPIHT [5] and JPEG 2000 [6]. For JPEG 2000, we do not consider image tiling since it degrades the image quality a lot. The results for JPEG 2000 have been obtained using Jasper [7], an official implementation included in the ISO/IEC 15444-5 standard. All of them use the same wavelet filter bank (Daubechies' B7/9) and have been written and compiled with the same level of optimization. In our comparison, we will use the standard images Lena and Barbara (monochrome, 8bpp, 512x512), and the larger and less blurred images Café and Woman (monochrome, 8bpp, 2560x2048, equiv. 5-Megapixel), from the

JPEG 2000 testbed. For more tests, the reader can download an implementation of the coder at the authors' web site <http://www.disca.upv.es/joliver/LowMemRLW>.

Table 1 shows a compression comparison for the evaluated images and coders. In general, our proposal performs as well as SPIHT does for less detailed images (Lena and Woman) and better than it for more complex images (Barbara and Café). It is due to the fact that SPIHT is based on coefficients trees, and fewer trees can be established in images with many details. On the contrary, JPEG 2000 is more efficient than our proposal in highly detailed images, since it defines more contexts and uses R/D optimization. However, our coder and JPEG 2000 are similar in low detailed images.

**Table 1.** PSNR (dB) with different bit rates and coders for the evaluated images. The numbers in parenthesis for our proposal correspond to the decrease of performance if the R/D improvements discussed in subsection 3.2 are not applied.

Lena (512x512)				Barbara (512x512)			
Codec \ rate	SPIHT	Jasper/ JP2K	Proposed Run Length	SPIHT	Jasper/ JP2K	Proposed Run Length	Run
1	40.41	40.31	40.37 (-0.14)	36.41	37.11	36.82 (-0.35)	
0.5	37.21	37.22	37.15 (-0.10)	31.39	32.14	31.90 (-0.29)	
0.25	34.11	34.04	34.03 (-0.08)	27.58	28.34	28.12 (-0.22)	
0.125	31.10	30.84	30.97 (-0.04)	24.86	25.25	25.19 (-0.08)	
Woman (2560x2048)				Café (2560x2048)			
Codec \ rate	SPIHT	Jasper/ JP2K	Proposed Run Length	SPIHT	Jasper/ JP2K	Proposed Run Length	Run
1	38.28	38.43	38.49 (-0.21)	31.74	32.04	31.89 (-0.26)	
0.5	33.59	33.63	33.72 (-0.15)	26.49	26.80	26.67 (-0.16)	
0.25	29.95	29.98	30.04 (-0.08)	23.03	23.12	23.10 (-0.12)	
0.125	27.33	27.33	27.40 (-0.04)	20.67	20.74	20.67 (-0.06)	

**Table 2.** Total memory required (in KB) to encode the Woman image with the compared algorithms. The numbers in parenthesis correspond to the memory that is saved if the R/D improvements are not used (it can be applied in both columns of our proposed algorithm).

Codec \ rate	Compressed Image File	SPIHT	Jasper/ JP2K	Proposed Run Length	Proposed with bit-stream in memory
1	640	42,888	62,768	1,256	1,896 (-180)
0.5	320	35,700	62,240	1,192	1,512 (-180)
0.25	160	31,732	61,964	1,192	1,352 (-180)
0.125	80	28,880	61,964	1,176	1,256 (-180)

The comparison in which our encoder clearly outperforms both SPIHT and JPEG 2000 is in memory consumption. Table 2 shows that, for a 5-Megapixel image, our proposal requires between 25 and 40 times less memory than SPIHT, and more than 45 times less memory than Jasper/JPEG 2000. In this table, the last column refers to the case in which the complete bitstream (i.e., the compressed image) is kept in memory while it is generated. Due to the computation order in the proposed wavelet transform, the coefficients from different subband levels are interleaved. Thus, instead of a single bitstream, we generate a different bitstream for every level. These different

streams can be kept in memory or saved in secondary storage. In addition, having a different bitstream for each level eases the decompression process, since the order in the inverse transform is just the reverse of the order in the forward one.

In this table, the memory estimated for executing a single process is about 650 KB. Hence, we can consider that the remaining memory is the data memory. Moreover, for our RLW coder, 180 KB can be saved if we use 8 lines per buffer instead of 16.

Since JPEG 2000 has more contexts and uses R/D optimization, it is more complex than our proposal. SPIHT is also more complex because it performs several image scans handling a different bit-plane each scan. Moreover, in cache-based systems, the proposed DWT makes better use of the cache. The last table shows an execution time comparison for two image sizes. Due to the former reasons, our algorithm clearly outperforms Jasper/JPEG 2000, and it is several times faster than SPIHT. In addition, we can speed it up in about 30% if no compression improvements are performed.

**Table 3.** Execution time (in Million of CPU Cycles) needed to encode images of different size. The numbers in parenthesis correspond to time reduction if no R/D improvements are applied.

Codec \ rate	Woman (2560x2048)			Lena (512x512)		
	SPHIT	Jasper / JP2K	Proposed Run Length	SPHIT	Jasper / JP2K	Proposed Run Length
1	3,669	23,974	1,855 (-587)	147	750	98 (-28)
0.5	2,470	23,864	1,291 (-377)	97	734	65 (-21)
0.25	1,939	23,616	970 (-259)	73	726	44 (-11)
0.125	1,651	23,563	783 (-197)	60	717	34 (-7)

## 5 Conclusions

In this paper, a wavelet image coder with state-of-the-art compression performance has been presented. The main contribution of this image coder is that it requires much less memory to work and thus, it is a good candidate for many embedded systems and other memory-constrained environments (such as digital cameras and PDAs). In addition, it is also several times faster than the other evaluated wavelet image coders.

## References

1. S. Mallat: A theory for multiresolution signal decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence, July 1989
2. M. Vishwanath: The recursive pyramid algorithm for the discrete wavelet transform. IEEE Transactions on Signal Processing, March 1994
3. C. Chrysafis, and A. Ortega: Line-based, reduced memory, wavelet image compression. IEEE Transactions on Image Processing, March 2000
4. J.Oliver, M.P.Malumbres: A fast wavelet transform for image coding with low memory consumption. 24<sup>th</sup> Picture Coding Symposium, December 2004
5. A. Said, A. Pearlman: A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans. on Circuits and Systems for Video Technology, June 1996
6. ISO/IEC 15444-1: JPEG 2000 image coding system, 2000
7. M. Adams: Jasper software reference manual. ISO 1/SC 29/WG 1 N 2415, October 2002

Part V

# Face Recognition

# Multiple Face Detection at Different Resolutions for Perceptual User Interfaces

Modesto Castrillón-Santana, Javier Lorenzo-Navarro, Oscar Déniz-Suárez,  
José Isern-González, and Antonio Falcón-Martel

IUSIANI

Edif. Ctral. del Parque Científico Tecnológico  
Universidad de Las Palmas de Gran Canaria, Spain  
mcastrillon@iusiani.ulpgc.es

**Abstract.** This paper describes in detail a real-time multiple face detection system for video streams. The system adds to the good performance provided by a window shift approach, the combination of different cues available in video streams due to temporal coherence. The results achieved by this combined solution outperform the basic face detector obtaining a 98% success rate for around 27000 images, providing additionally eye detection and a relation between the successive detections in time by means of detection threads.

## 1 Introduction

People detection is a basic ability to be included in any Vision Based Interface [14] in order to use computer vision technology to perceive the user in a Human Computer Interaction (HCI) context. Among the different approaches for this purpose, face detection has been a revisited topic in the recent literature.

The face detection problem, defined as: *to determine any face -if any- in the image returning the location and extent of each* [18], seems to be solved, according to some recent works [9, 11, 16]. Particularly for video stream processing, these approaches focus the problem in a monolithic fashion, forgetting elements that the human system employs: temporal and contextual information, and cue combination.

The work presented in this paper describes a real-time vision system which goes beyond traditional still image face detectors. The resulting system is an approach for robust multiresolution real-time multiple face detection which combines different cues based on an obvious connection that exists between frames, i. e. temporal coherence. The resulting approach achieves better detection rates for video stream processing and cheaper processing costs than outstanding and public available face detection systems.

### 1.1 Previous Work

Face detection methods are classified according to different criteria as recent face detection surveys do [5, 18]. In our opinion these techniques can be classified into two main families according to the information used to model faces:

- Pattern based or Implicit: These approaches work by searching exhaustively a previously learned pattern at every position and different scales of the whole input image.
- Knowledge based or Explicit: These approaches increase processing speed by taking into account face knowledge explicitly, exploiting and combining cues such as color, motion, face and facial features geometry, and appearance.

Recent window shift based approaches, i.e. pattern based, have achieved impressive results applied even to video streams [9, 11, 16]. However, the exclusive use of a monolithic approach has the disadvantage of despising a main cue useful for video processing: temporal coherence. Any face detected in a frame provides valid information which can be used to speed up the process in the next frames.

## 2 The Face Detection Approach

Our approach is related to both categories described in the previous section, as it makes use of both implicit and explicit knowledge to get the best of each one. The explicit knowledge is based on the face geometry and the descriptors extracted from a detection: color and appearance. On the other side, the implicit knowledge is integrated using the general object detection framework integrated in the Open Computer Vision Library (OpenCV) [6]. This framework is based on the idea of a boosted cascade classifier [16] but extends the original feature set and provides different boosting variants for learning [10]. The framework combines increasingly more complex classifiers in a cascade, allowing background regions of the image to be quickly discarded while spending more time on promising object-like regions.

The face detection approach here described has two different working modes depending on recent face detection events reported:

**After no detection:** This working mode takes place at the beginning of an interaction session, when all the individuals are gone from the field of view, or if nobody is detected for a while. The approach basically makes use of two window shift detectors based on the general object detection framework described in [16]. These two brute force detectors, integrated in the last OpenCV release [6], are the frontal face detector described in that paper, and the local context based face detector described in [8]. The last one achieves better recognition rates for low resolution images if the head and shoulders are visible. The respective minimum size searched are  $24 \times 24$  and  $20 \times 20$  pixels. In order not to waste processing time, the detectors are executed alternatively.

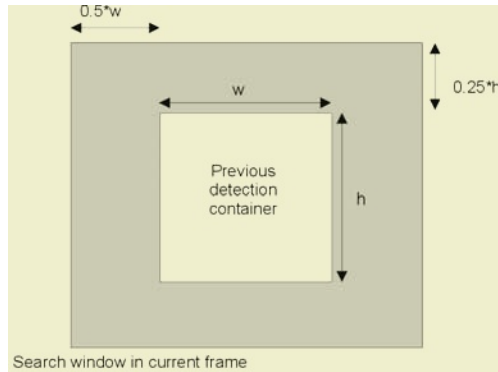
For any face detected, the system tries to detect its facial features assuming that it is a frontal face, and therefore its facial features would verify some geometric restrictions. The current implementation searches only the eyes, using a process similar to the one employed in [1] just for a single face detection approach. It was however improved by the addition of different alternatives for eye detection as described below:

1. *Skin blob detection*: Once a face is detected, its skin color is modelled using red-green normalized color space [17], considering just the center of the face container provided by any of the Viola-Jones based detectors. The system heuristically removes elements that are not part of the face, e.g. neck, and fits an ellipse to the blob in order to rotate it to a vertical position [12].
2. *Eyes location*: At this point, the approach searches eye candidates in the likely areas inside the skin blob considering that the face detected is a frontal face. Different candidate pairs are checked for their appearance until one of them, is accepted. The cues used for this purpose are:
  - (a) *Dark areas*: Eyes are particularly darker than their surroundings [2].
  - (b) *Viola-Jones based eye detector*: As the eye position can be roughly estimated and therefore restricted, a Viola-Jones based eye detector provides very fast results. The detector searches eyes with a minimum size of  $16 \times 12$  pixels. For small faces, they are scaled up before performing the search.
  - (c) *Viola-Jones based eye pair detector*: If other cues fail, the eye pair detection can provide another estimation for eye positions. The minimum pattern size searched is  $34 \times 8$ .
3. *Normalization*: Eye positions, if detected, provide a measure to normalize the frontal face candidate. The normalization step allows further face processing modules to reduce the problem dimensionality.
4. *Pattern Matching Confirmation*: Once the likely face has been normalized, its appearance is checked in two steps making use of Principal Component Analysis (PCA) spaces [7]. The PCA spaces were built using a face dataset of 4000 facial images extracted from internet and annotated by hand.
  - (a) *Eye appearance test*: A certain area ( $11 \times 11$ ) around both eyes in the normalized image is projected to a PCA space and reconstructed. The reconstruction error [4] provides a measure of its eye appearance, and can be used to identify incorrect eye detections.
  - (b) *Face appearance test*: A final appearance test applied to the whole normalized image. The image is first projected to a PCA space, and later its appearance is tested using a Support Vector Machine (SVM) classifier [15].

**After recent detection(s)**: As briefly mentioned above, for each detected face, the system stores not only its position and size, but also its average color using red-green normalized color space [17], and the patterns of the eyes (if detected) and the whole face. Thus, a face is characterized by  $f = \langle pos, size, red, green, leye_{pos}, leye_{pattern}, reye_{pos}, reye_{pattern}, face_{pattern} \rangle$ . These features direct different cues in the next frames which are applied opportunisticly in an order based on their computational cost and reliability.

- *Eye tracking*: A fast tracking algorithm [3] is applied in an area that surrounds previously detected eyes, if available. The tracker makes use of a fixed pattern size for both eyes,  $24 \times 24$ , and searches the minimum difference in the search area as follows:





**Fig. 1.** The search area used for each detected face in the next frame is defined as an expansion of the previous face detection container.

$$D(u, v) = \sum_{Area} |I(u + i, v + j) - P(i, j)| \quad (1)$$

Eye patterns are previously saved with the first detection, and updated according to the strategies described in [3], i.e. only if there is a notorious change in relation to the original pattern, and this difference could confuse the tracker with any other pattern of the close context. If the difference reported is too big, the pattern will be considered lost.

- Basic face detector: The Viola-Jones face detector [16] searches faces but only in an area that covers the previous detection, see Figure 1. This strategy significantly reduces processing time.
- Local context face detector: If previous techniques fail, the local context based face detector is applied in an area that includes the previous detection [8], see Figure 1.
- Skin color: The integration of other cues, likely weaker, help to improve the final system performance and robustness. Skin color based approaches for face detection have the lack of robustness for different conditions. A well known problem is the absence of a general skin color representation for any kind of light source and camera [13]. However, the skin color extracted from the face previously detected by the Viola detector can be used to estimate facial features position by means of the color blob, as described above. If previous cues fail, the modelled skin color is used to locate the face, and therefore it is searched in the window that contains the previous detection, see Figure 1. The new sizes and positions are coherently checked, due to the fact that the skin color container is not allowed to experiment large size changes just to avoid an incorrect color updating mechanism.
- Face tracking: If everything else fails, the prerecorded face pattern is searched in an area that covers previous detection [3], see Figure 1. The tracking pattern has a fixed size, for that reason the system scales down the face to fit it in the pattern size. The scale ratio is stored and later

used if necessary to scale down the search area in the next frame. This action helps reducing the tracking shift problem. However, the tracking is not allowed to be the only valid cue for more than some consecutive frames in order to avoid tracking problems. Instead, the other cues should confirm the human presence, from time to time, or the person will be considered lost.

For each previous detection, these techniques are applied until one of them finds a new face coherent with the previous detection. Whenever a face is detected, and its eyes were not tracked, the skin color is used for facial features detection as explained above for the *After no detection* working mode. Also, every third frame one of the Viola-Jones based detectors is applied to the whole image in order to detect new faces. Those new faces are compared with those already detected by temporal coherence and those which are redundant removed. If no faces are detected for a while, the process switches to the default *After no detection* working mode.

The approach described considers the possibility of multiple face detection, as no restriction is imposed in that sense. It is interesting to relate the detection information achieved in the consecutive frames, especially when multiple individuals are present. During the video stream processing, the face detector gathers a set of detection threads,  $IS = \{dt_1, dt_2, \dots, dt_n\}$ . A detection thread contains a set of continuous detections, i.e. detections which take place in different frames but are related by the system in terms of position, size and pattern matching techniques. Thus, for each detection thread, the face detector system provides a number of facial samples,  $dt_p = \{x_1, \dots, x_{m_p}\}$ , which correspond to those detections for which also the eyes were located.

The Viola-Jones based detectors have some level of false detections. For that reason a new detection thread is created only if the eyes have been also detected. The use of color and tracking cues after a recent detection is reserved to detections which are already considered part of a detection thread. In this way, spurious detections do not launch cues which are not robust enough, in the sense that they are not able to recover from a false face detection.

Ideally a detection thread contains samples detected from a single individual. However, different detection threads can correspond to the same individual, aspect which is not checked by the current implementation. Gaps are allowed during detection thread life, but a detection thread is considered lost if after a predefined number of frames it is not correctly associated to a new detection.

### 3 Performance Results

For static images the approach provides a performance which combines the results achieved for the standard Viola-Jones face detector [16] and the local context based face detector [8]. We refer the reader to those works to get precise information for static images results.



**Fig. 2.** Different samples of some sequences.



**Fig. 3.** From left to right: 1) Both faces and their eyes are detected, 2) the face on the right is detected by tracking the face pattern due to the Viola based detectors failure, 3) the left face is detected using skin color and the right one by means of the local context face detector, 4) the same for the left face, the right one is found by tracking, 5) face pattern tracking is not allowed to be the only valid cue for many consecutive frames, so the right face detection thread is considered missed, and 6) the right face recovers its vertical position and it is fused with the latent detection thread.

The strength of our approach is exploited in video stream processing thanks to cue integration. 70 sequences, see Figure 2, corresponding to different individuals, cameras and environments with a resolution of  $320 \times 240$  were recorded and processed. The total set contains 27271 images, presenting all of them a face easily detected by a human. The average processing time of 60 msecs. using a PIV 2.2Ghz, allowed the system to associate 26875 (98.5%) detections to a detection thread, see Figure 3. As described in that figure, some of those detections are not provided by the Viola-Jones based detectors, but by the cue integration approach. From those detections, their eyes were also located in 70% of them. It must be observed that eyes are located only for frontal poses in the current implementation.

At least 10 of those sequences reported detections which correspond to non face patterns. These detections were correctly not assigned to any detection thread as the eyes were not found and their position, color and size were not coherent with any active detection thread.

Only for 3 (4%) sequences with a single individual, the detection thread was not unique. In these sequences this was due to the fact that at a certain point a detection thread was incorrectly fused with an erroneous detection provided by the Viola-Jones based detectors. However, in all the cases the detection thread was shortly considered lost, and therefore some frames later the still present face was newly detected, and a new detection thread created.

For single individuals sequences this is an impressive result considering the large changes in pose experimented in many of the sequences. The processing rates achieved make the system suitable for further processing in the field of perceptual user interfaces.

For multiple individuals sequences, the system needs more time as more faces are tracked simultaneously, in our experiments the processing time is increased

around 20 msec. per. This effect can be reduced by decreasing the number of times per second that new faces are searched in the whole image, due to the fact that two faces cover more area and therefore it is less likely the presence of a new face. It must also be noticed that in these sequences as no appearance cue is used to relate a detection in the next frame with a previous one, the system is not currently able to manage coherently a situation when different detection threads can overlap, i.e., there is occlusion. It is not sure that after the occlusion between two individuals, the detection threads will be properly assigned to the new detections.

## 4 Conclusions

We have described a system which combines multiple cues taking into account their respective computational cost and reliability in the problem of face detection. The approach developed provides fast multiple face detection at different resolutions for standard webcam images, i.e.  $320 \times 240$ , suitable for perceptual user interfaces.

The system is also able to provide information about the relation of the detections in time, reporting good results in the experiments. Currently detection threads can contain among their samples some with bad eye detections, particularly when the face is not completely frontal. In this sense the appearance test must be improved. However, the system is always able to recover once a frontal face is present. Future work must cover the detection of other facial elements in order to have a more robust facial features detection for non frontal poses, as in the current implementation it is only based on eye detection.

Another interesting step to be done is the integration of additional descriptors, e.g. identity, t-shirts color, etc., in order to be able to manage situations with occlusions between individuals, which right now are not specifically analyzed.

## Acknowledgments

Work partially funded by research projects of the Univ. of Las Palmas de Gran Canaria UNI2003/06, Canary Islands Autonomous Government PI2003/160 and PI2003/165 and the Spanish Ministry of Education and Science and FEDER funds (TIN2004-07087).

## References

1. M. Castrillón Santana, F.M. Hernández Tejera, and J. Cabrera Gámez. Encara: real-time detection of frontal faces. In *International Conference on Image Processing*, Barcelona, Spain, September 2003.
2. Stefan Feyrer and Andreas Zell. Detection, tracking and pursuit of humans with autonomous mobile robot. In *Proc. of International Conference on Intelligent Robots and Systems, Kyongju, Korea*, pages 864–869, 1999.

3. Cayetano Guerra Artal. *Contribuciones al seguimiento visual precategórico*. PhD thesis, Universidad de Las Palmas de Gran Canaria, Octubre 2002.
4. Erik Hjelmas and Ivar Farup. Experimental comparison of face/non-face classifiers. In *Procs. of the Third International Conference on Audio- and Video-Based Person Authentication. Lecture Notes in Computer Science 2091*, 2001.
5. Erik Hjelmas and Boon Kee Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3), 2001.
6. Intel. Intel open source computer vision library, b4.0. [www.intel.com/research/mrl/research/opencv](http://www.intel.com/research/mrl/research/opencv), August 2004.
7. Y. Kirby and L. Sirovich. Application of the karhunen-loève procedure for the characterization of human faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1), July 1990.
8. Hannes Kruppa, Modesto Castrillón Santana, and Bernt Schiele. Fast and robust face finding via local context. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, October 2003.
9. Stan Z. Li, Long Zhu, ZhenQiu Zhang, Andrew Blake, HongJiag Zhang, and Harry Shum. Statistical learning of multi-view face detection. In *European Conference Computer Vision*, 2002.
10. Rainer Lienhart, Luhong Lian, and Alexander Kuranov. An extended set of haar-like features for rapid object detection. Technical report, Intel Research, June 2002.
11. Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.
12. Karin Sobottka and Ioannis Pitas. A novel method for automatic face segmentation, face feature extraction and tracking. *Signal Processing: Image Communication*, 12(3), 1998.
13. Moritz Storrang, Hans J. Andersen, and Erik Granum. Physics-based modelling of human skin colour under mixed illuminants. *Robotics and Autonomous Systems*, 2001.
14. M. Turk. Computer vision in the interface. *Communications of the ACM*, 47(1):61–67, January 2004.
15. V. Vapnik. *The nature of statistical learning theory*. Springer, New York, 1995.
16. Paul Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, 2001.
17. Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfunder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7), July 1997.
18. Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

# Removing Shadows from Face Images Using ICA<sup>\*</sup>

Jun Liu, Xiangsheng Huang, and Yangsheng Wang

CASIA-SAIT HCI Joint Lab, Institute of Automation, Chinese Academy of Sciences  
jliu@hci.ia.ac.cn

**Abstract.** Shadows produce troublesome effects in many computer vision applications. The idea behind most current shadow removal approaches is locating shadows and then removing them[1][4]. However, distinguishing shadow edges due to shadows from reflectance edges due to reflectance changes is a difficult problem, particularly in a single image. In this paper, we focus on the shadow removal problem in face recognition, and take a novel method based on ICA (Independent Component Analysis) to remove shadows from a single face images. The training set contains face images without shadows. Firstly, we applied derivative filters on training images to derive face edge maps, and then perform ICA on filtered training set to construct pixel ICA subspaces which can be used to remove shadow edges from the filtered versions of a single test image. After the shadow edges removal process, a shadow free image can be reconstructed using an approach similar to [7]. Unlike previous shadow removal approaches, our method can remove shadows from a single gray image. Experimental results demonstrate that the proposed approach can effectively eliminate the effects of shadows in face recognition.

## 1 Introduction

Shadows in images may cause problems to many algorithms in the fields of image processing and computer vision, such as object detection and recognition, and removing them can greatly improve the results of these algorithms. One possible solution to the shadow removal problem can be locating shadows and removing them which generally requires identification and removal of shadow edges. However, the disambiguation of shadow edges due to shadows and reflectance edges due to reflectance changes is a difficult problem and has a long history in computer vision research [5].

Recently, a method to derive shadow free images was presented by Weiss [7]. The example used in the paper was a sequence of grey-scaled images captured with a stationary camera over a period of time such that the illumination in the scene (specifically the position of the shadows) changes considerably. Assuming that reflectance changes are constant in the scene and that shadows move in

---

<sup>\*</sup> This work supported by National Natural Science Foundation of China, Project Number: 60473047.

the image sequence, it follows that the median edge map of the image sequence can be used to remove shadow edges, while reserving reflectance edges. Given the reflectance edge map, a shadow free image that depends only on reflectance can be recovered. The same method is introduced in [8] to handle shadows in an intelligent transportation system. It is proven to perform quite well for shadow removal, however its applicability is limited since it requires a sequence of images of the same scene in which only the illumination varies with time.

Removing shadow directly from a single image is a challenging task. In [3], Finlayson assumed that camera sensor sensitivities behave like delta functions and that lighting in the scene can be approximated by Planckian lights. These constraints ensures the extraction of 1D-illuminant invariant image without shadows from a single RGB image. Then applying edge-detection on the invariant image and the original image, and removing the edges that exist in the original image but not in the invariant image, a reflectance edge map can be derived. Similar to Weiss’s method, reintegrating the edge map can result in a shadow free image. This method outperforms the previous approaches in some aspects. However, its assumptions can’t be exactly met in real scenes. So the shadow removal effect in real images aren’t perfect. Moreover, It can’t deal with gray images.

When thinking about how we might remove shadows from an image it is important that we consider the applications we are interested in, because this can have a bearing on what restrictions we place on how we solve the problem. In this paper, we focus on the shadow removal problem in the face recognition and the restriction is we handle with a class of object. Faces without shadows have similar edges, and a global constraint can be derived from the filtered face images to help remove shadow edges. We chose ICA factorial code architecture to train the global models because the face edge maps are sparse, and we need to deal with them pixelwisely. The training set contains face images without shadows. Our work began with applying derivatives filters on the training images, and then ICA was performed on the filtered training images according to each filters to obtain the global models separately. Given a filtered test face image, we found that a minimum squared error approximation of it derived with the global models was shadow edge free. And then reintegrating the edge maps with shadow edges removed using an approach similar to [7] resulted in a shadow free image. Experimental results showed that our proposed method can effectively remove shadows from a single gray face image.

The remainder of this paper is organized as follows. Section 2 recapitulates the related work of Weiss. Section 3 introduces our novel method to remove shadows from a single image using independent component analysis. Experiments and results are reported in Section 4. Section 5 concludes this paper.

## 2 Related Work and Discussion

Weiss’s method works in the framework of “intrinsic image”, which was introduced by Barrow and Tenenbaum in 1978. According to their definition, an image  $I(x, y)$  can be decomposed into a product of two images:

$$I(x, y) = L(x, y)R(x, y) \quad (1)$$

where  $L(x, y)$  is an illumination image and  $R(x, y)$  is a reflectance image which is shadow free.

Obviously, recovering two intrinsic images from a single image is an ill-posed problem: the number of unknowns is twice the number of equations. However, deriving intrinsic images from a sequence of images is a relatively easier problem. Given a sequence of  $T$  images  $\{I(x, y, t)\}_{t=1}^T$ , a single reflection image  $R(x, y)$  and  $T$  illumination images  $L(x, y, t)$  can be derived.

For convenience, This method works in the log domain.

$$\begin{aligned} \log I(x, y, t) &= \log L(x, y, t) + \log R(x, y) \\ i(x, y, t) &= l(x, y, t) + r(x, y) \end{aligned} \quad (2)$$

Previous researches showed that when derivative filters are applied to natural images, the filter outputs tend to be sparse. So when a horizontal derivative filter  $f_x$  and a vertical derivative filter  $f_y$  are applied on  $l(x, y, t)$ , the resulting filter outputs  $l_n(x, y, t) : l_x, l_y$  are sparse and then can be approximated with a Laplacian distribution  $P(l_n) = \frac{1}{Z} e^{-\alpha|l_n|}$ . Applying derivative filters  $\{f_n : f_x, f_y\}$  on  $i(x, y, t)$ , we have

$$i(x, y, t) \star f_n = l(x, y, t) \star f_n + r(x, y) \star f_n \quad (3)$$

$$i_n(x, y, t) = l_n(x, y, t) + r_n(x, y) \quad (4)$$

Then the ML estimate of filtered reflectance image  $\hat{r}_n(x, y)$  are given by:

$$\hat{r}_n(x, y) = \text{median}_t i_n(x, y, t) \quad (5)$$

Then  $r$  can be recovered via solving the overconstrained systems of linear equations:

$$\hat{r} \star f_n = \hat{r}_n \quad (6)$$

In this case, the solution can be given by:

$$\hat{r} = g \star [f_x(-x, -y) \star r_x + f_y(-x, -y) \star r_y] \quad (7)$$

where  $f_n(-x, -y)$  is a reversed copy of  $f_n(x, y)$ , and  $g$  is the solution of

$$g \star [f_x(-x, -y) \star f_x(x, y) + f_y(-x, -y) \star f_y(x, y)] = \delta \quad (8)$$

Note that  $g$  is independent of the image sequence, so can be computed offline. The whole process is illustrated in Figure 1.

The prerequisites of this method is a sequence of images of one fixed scene only with significantly changing illuminations. However, it is difficult to meet in face recognition. Acquiring strictly aligned face images with illumination changes needs face alignment technologies insensitive to illumination. But all of current face alignment algorithms will be affected dramatically by illumination changes. So exploring methods to remove shadows from just a single face image is critical to face recognition.



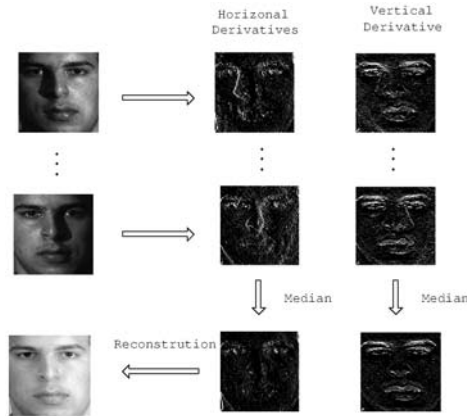


Fig. 1. Illustration of the weiss's algorithm.

Let's look at equation (4), what we need is an estimation of  $r_n(x, y)$ . As described in [6], face images under arbitrary illuminations construct a low-dimension subspace. Our motivation is to construct filtered face illumination subspaces containing filtered face images without shadow edges. Given a face image with shadows, we can project its filtered versions into these subspaces to produce edge maps with shadow edges removed. This eliminates the need to locate shadows in images, instead we introduce the global constraint of face images into the shadow removal process. Because the filtered faces are sparse and we treat each of them as an observation, so the factorial code architecture of ICA best fits to construct these subspaces. After removing shadow edges from the filtered face images, a reconstruction process can be applied to derive a shadow free image.

### 3 Removing Shadows from Face Images

#### 3.1 Independent Component Analysis

The goal of independent component analysis (ICA) is to decompose a set of observations into a basis whose components are statistically independent or, at least, are as independent as possible. Given  $N$  face images, prior to the application of ICA on the face images, each image has been scanned rowwise in order to form a column feature vector. All the feature vectors have been collected in a matrix  $X$  whose columns contain the images. Let us now suppose that each image (columns of  $X$ ) represents a linear combination of some underlying basis images. In the matrix form we can write  $X = AS$ , where  $A$  are the basis images associated with a set of independent coefficients vector (source) of  $S$  [2]. All we want to do is to estimate  $A$  by  $D^{-1}$ , where the unmixing matrix  $D$  is the learned ICA weight matrix, such that  $C_{train} = DX$  and  $C_{train} \approx S$ . Therefore, each column of  $C_{train}$  consists of the independent coefficients,  $C_{train}$ , for

the linear combination of basis images in  $A$  that comprised each face image  $x$ . Since ICA attempts to make  $C_{train}$  as independent as possible,  $C_{train}$  is called a factorial code for face images [2]. Note that, under this configuration the pixels are independent across the same image. That is, the coefficients  $c$  found in the columns of  $C_{train}$  are independent and not the basis images.

In order to have a controlled reduction of the number of independent components,  $m$  linear combinations of the original images, namely the first  $m$  PCA coefficients of the images, are chosen. ICA was performed on the matrix  $R_m^T$  whose columns are the PCA coefficients of the training images. Let  $P_m^T$  be the modal matrix where rows are the  $m$  principal eigenvectors. Matrix  $R_m^T$  is then given by  $R_m^T = P_m^T X$ . Hence:  $C_{train} = DR_m^T$ . Subsequently, a whitening process is applied to  $R_m^T$  to normalize the data. If the row means are subtracted from  $R_m^T$  and the resulting matrix is passed through a zero-phase whitening filter which is twice the inverse square root, the whitening transformation is written as  $W = 2(\frac{1}{N}R_m^T R_m^T)^{-\frac{1}{2}}$ . Therefore, the zero-mean input matrix can be decomposed as the product of the unmixing matrix and the whitening matrix  $D_w = DW$ . Accordingly, (1) is rewritten as  $C_{train} = D_w R_m^T$ . The unmixing matrix  $D_w$  must be learned by ICA during training. An iterative process for updating  $D_w$  yields the independent coefficients. Different approaches exist for this purpose. A typical one is maximum entropy method. Let  $c_{train,i}$  be the  $i$ -th column vector of  $C_{train}$ ,  $c_{train,i} = (c_{i1}, c_{i2}, \dots, c_{ij}, \dots, c_{im})^T$ , and  $g(\xi) = \frac{1}{1+e^{-\xi}}$  be a nonlinearity applied component wise to the elements of  $c_{train,i}$  to yield the vector  $z_i = g(c_{train,i})$ . An updating equation for  $D_w$  based on  $c_{train,i}$  at each iteration  $k$ , is given by :

$$D_w(k+1) = D_w(k) + \eta [I + (1 - 2z_i(k))c_{train,i}^T(k)]D_w(k) \quad (9)$$

where  $\eta$  is the learning rate,  $I$  is the identity matrix and  $1$  is a  $m \times 1$  of ones. Obviously,  $c_{train,i}(k) = D_w(k)r_{m,i}$  where  $r_{m,i}$  is the  $i$ -th column of  $R_m^T$  and  $z_i(k) = g(c_{train,i}(k))$ . Once we have finished training and obtained  $C_{train}$ , the coefficients for a test image  $x_{test}$  can be represented as

$$c_{test} = D_w P_m^T x_{test} \quad (10)$$

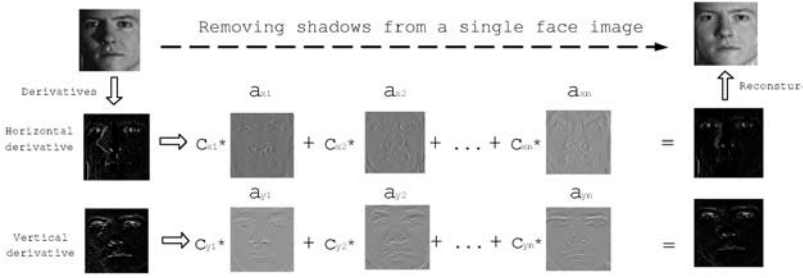
where  $x_{test}$  has zero mean.

### 3.2 Our New Method of Shadow Removal

According to the ICA factorial representation of face images described above, given a test image  $x_{test}$ , we have an image synthesis model as:

$$x_{syn} = AC_{test} \quad (11)$$

where the elements of  $C_{test}$  are given by equation (10). Figure 2 illustrates this image synthesis model. Note that during training we perform ICA on filtered



**Fig. 2.** Image synthesis model of ICA. The independent coefficients,  $c$ , for the linear combination of basis images in  $A$  synthesize each face image. The synthesized face edge map is shadow edge free.



**Fig. 3.** Examples of ICA basis images.

face images, not original face images because we need sparse input. The training process is given below.

Given training face images without shadows  $X = \{x_i | i = 1, \dots, n\}$ , and a set of derivative filters  $F = \{f_x, f_y\}$ , where  $f_x$  is a horizontal derivative filter and  $f_y$  is a vertical derivative filter, as those in section 2. Applying these derivative filters to the face images, we have two sets of face edge maps. Performing ICA on both sets separately, we get two sets of ICA basis images that make up of matrix  $A_x$  and  $A_y$ . Some examples of these basis images are shown in figure 3.

After the training, we have the unmixing matrix  $D$  and basis images contained in  $A$ . Using equation (11), two filtered versions of test images with shadows can be projected into the ICA subspaces separately, deriving two synthesized edge maps with shadow edges removed ( $\hat{r}_x$  and  $\hat{r}_y$ ) as illustrated in figure 2.

The reconstruction is given by:

$$x_{new} = g \star [f_x(-x, -y) \star \hat{r}_x + f_y(-x, -y) \star \hat{r}_y] \tag{12}$$

where  $f_n(-x, -y)$  is a reversed copy of  $f_n(x, y)$ , and  $g$  is the solution of equation (8).  $x_{new}$  is the shadow free image we derive from a single face image with shadows.



Fig. 4. Some examples of the CMU face database.

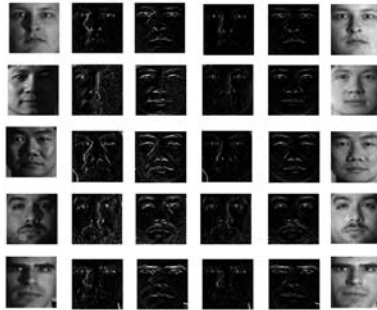


Fig. 5. Removing shadows from a single face image.

## 4 Experiments and Results

We have extensively test our method on a subset of the CMU PIE face image database. This database consists of 68 persons' face images captured under changing illuminations. We choose 20 images of each person under different illuminations, crop and scale them to  $100 \times 100$  to form the subset. Examples of this subset are shown in figure 4. Five face images without shadows of each person were chosen as the training set, others as the test set. Experimental results show that our method can successfully remove shadows from a single face image (figure 5). In figure 5, column 1 is the original test face image, the horizontal and vertical derivatives of which are shown in column 2 and 3, column 4 and 5 are the corresponding face edge map synthesized using the ICA models, from which we can see that shadow edges are gone. The last column shows the reconstructed shadow free image.

Experimental results show that our method can achieve good shadow removal effects, and we need only a single image. We further perform face recognition experiment on the test set. The test set is divided into 3 subsets, subset 1 contains  $68 \times 5$  images without shadows, subset 2 contains  $68 \times 5$  with small shadows, and subset 3 contains  $68 \times 5$  with large shadows. Direct correlation method is used to recognize the face images after shadow removal. We compared our method with raw method (without any preprocessing) and PCA method. The recognition results are shown in table 1. It is obvious that our method significantly improves the recognition rate on face images with shadows.

## 5 Conclusion

The performance of algorithms in computer vision decrease when there are shadows in images. Typical shadow removal approaches try to locate shadows in the

**Table 1.** Face Recognition Results on CMU PIE.

	Subset 1	Subset 2	Subset 3
Raw Method	94.2%	61.5%	45.3%
PCA Method	96.3%	78.2%	69.9%
Our Method	97.3%	94.5%	92.1%

image, which, however, is a difficult task. In this paper, we present an effective ICA-based method to remove shadows from a single face image. Firstly we remove shadow edges from face edge maps of the test image using ICA learning method, and then reconstruct a shadow free image from the face edge maps with shadow edges removed. Our novelty is that we introduce the global constraints of face images into the shadow removal process using learning method, which enable us to remove shadows from a single gray image. The experimental results showed that using prior learning our method can effectively improve the performance of face recognition on face images with shadows.

## References

1. Kobus Barnard and Graham Finlayson. *Shadow Identification using Colour Ratios*. Proceedings of the Eighth Color Imaging Conference: Color Science, Systems and Applications, pp. 97-101, 2002.
2. Movellan J.R. Bartlett, M.S. and T.J. Sejnowski. *Face recognition by independent component analysis*. IEEE Transactions on Neural Networks 13(6) p. 1450-64, 2002.
3. Hordley S.D G.D.Finlayson and Drew M.S. *Removing shadows from images*. ECCV02, pages 823-836, 2002.
4. Mark S. Drew Graham D. Finlayson, Steven D. Hordley. *Removing Shadows From Images using Retinex*. Color Imaging Conference 2002: 73-79.
5. A.D. Jepson R. Gershon and J.K. Tsotsos. *Ambient illumination and the determination of material changes*. J. Opt. Soc. Am. A, 3:1700-1707, 1986.
6. R.Basri and D.Jacobs. *Lambertian reflectance and linear subspaces*. Proc. ICCV, 2001.
7. Y. Weiss. *Deriving intrinsic images from image sequences*. Proc. of 9th IEEE Int'l Conf. on Computer Vision, pp. 68-75, Jul., 2001.
8. Katsushi Ikeuchi Yasuyuki Matsushita, Ko Nishino and Masao Sakauchi. *Illumination Normalization with Time-dependent Intrinsic Images for Video Surveillance*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2003)v, Vol.1, pp.3-10, June 2003.

# An Analysis of Facial Description in Static Images and Video Streams

Modesto Castrillón-Santana, Javier Lorenzo-Navarro,  
Daniel Hernández-Sosa, and Yeray Rodríguez-Domínguez

IUSIANI

Edif. Ctral. del Parque Científico Tecnológico  
Universidad de Las Palmas de Gran Canaria, Spain  
mcastrillon@iusiani.ulpgc.es

**Abstract.** This paper describes an analysis performed for facial description in static images and video streams. The still image context is first analyzed in order to decide the optimal classifier configuration for each problem: gender recognition, race classification, and glasses and moustache presence. These results are later applied to significant samples which are automatically extracted in real-time from video streams achieving promising results in the facial description of 70 individuals by means of gender, race and the presence of glasses and moustache.

## 1 Introduction

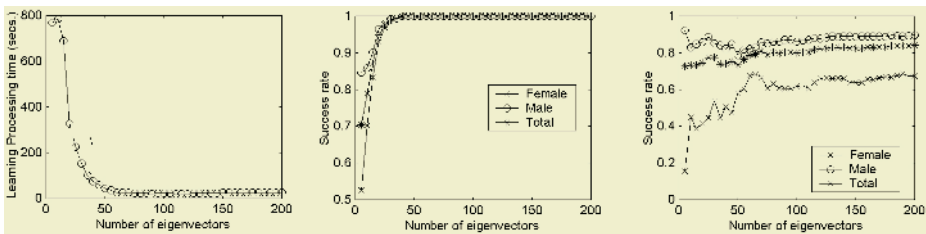
Human beings are sociable by nature and use their sensorial and motor capabilities to communicate with their environment. If Human to Computer Interaction (HCI) were more similar to human to human communication, accessing HCI devices would be easier and this fact would improve their social acceptability, becoming non-intrusive, more natural and comfortable [9].

Among the different channels used for human communication, the face has great importance conveying to humans a wealth of social signals, being therefore considered the center of human communication [6]. They tell us who the person is, or help us to guess features that are interesting for social interaction such as gender, age, expression and more. That ability allows us to react differently with a person based on the information extracted visually from his/her face. For these and other reasons, computer-based facial analysis is becoming widespread, covering applications such as identity recognition, gender recognition, facial expression analysis, etc.

The contribution of this work is the analysis of an appearance based approach for semantic facial description of individuals in static images and during an interactive session. The paper is organized as follows: in Section 2 the approach used for facial description in still images is described and tuned for the problems selected. Section 3 considers the application to video streams, establishing a criteria for pattern selection during interaction. Finally, in Section 4 the main conclusions of the work are outlined, as well as directions for future development.

## 2 Facial Description

The facial descriptors considered in this work are: gender, race, and the presence or not of moustache and glasses. In the literature different works have tackled the problem of gender recognition. A recent approach based on principal Components Analysis (PCA) achieves high performance also for low resolution images [8]. In [7] a Gabor wavelet representation on selected points is used with good results in gender and race classification. There are different references [4, 12] which try to detect the presence of glasses in a face, but we have not found any reference tackling the the problem of the moustache presence.

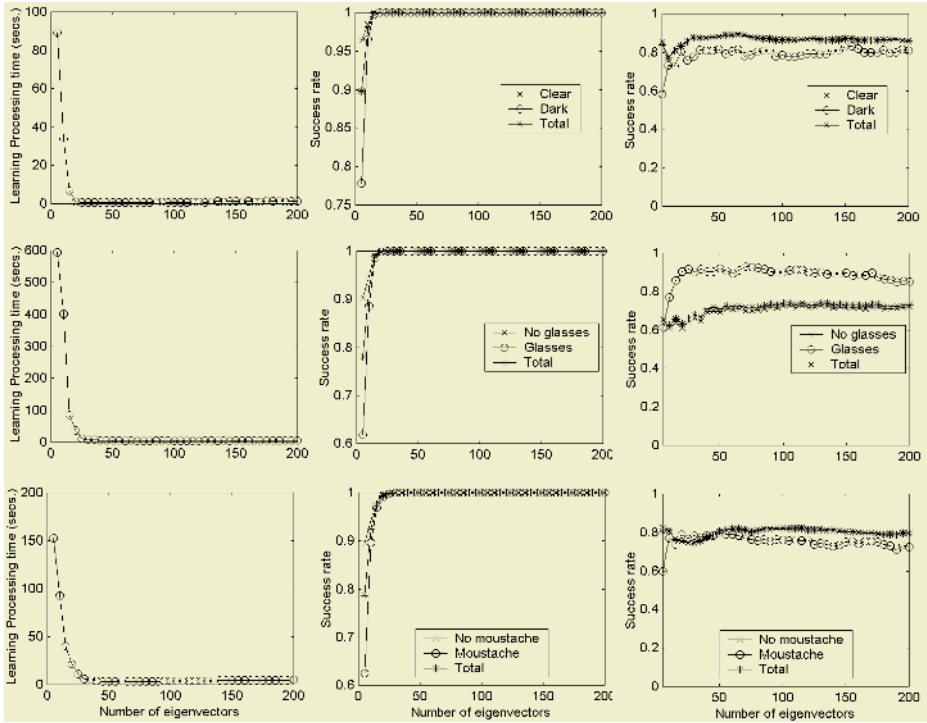


**Fig. 1.** Gender results: left) Model training time, middle) success rates for training set, and right) success rate for test set.

To tackle the problem, a representation mechanism must first be established to represent faces once the input data, i.e. the images, are available. It is interesting to reduce the data dimensionality to encode the face image without losing information. We selected a well known face representation space in advance: the PCA space due to its economical advantages [5]. The different classifications are performed in that representation space by means of a Support Vector Machines (SVM) classifier [10]. This combination *PCA + SVM* has been chosen for being well known by the community and the good performance results achieved [2].

The different classifiers performance is analyzed in relation to the number of eigenvalues used for representation, in order to get the best number for reliable classification in each problem. To define the PCA space, we have previously annotated the eye positions of 6000 faces of different people taken from internet. These images have been normalized according to eye positions obtaining  $59 \times 65$  samples which were used for the gender and race descriptors, and more localized areas to check the presence of glasses and moustache, see Figure 4.A. The PCA space calculation using 4000 of them required 12 hours in a PIV 2.2 Ghz. Different training and test sets have been set up for each problem, see Table 1.

**Gender recognition:** The results, see Figure 1, show that the training set needs around 40 – 50 eigenvalues to be perfectly classified, while the test set presents a balanced improvement for both sets up to 70 eigenvalues. The required training time is also reduced for more than 60 eigenvalues.



**Fig. 2.** Each row represents the ratios achieved for the different descriptors: gender, skin, glasses and moustache. In each row left) Model training time, middle) success rates for training set, and right) success rate for test set.

According to some results on human perception [3], the experiments have been also performed considering only the eyes area, achieving a performance only 5 points lower.

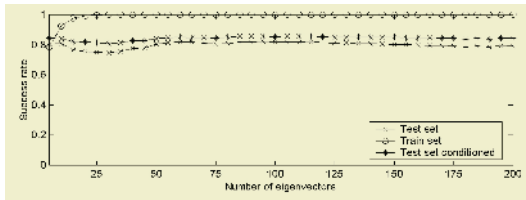
**Race classification:** Restricted by the face database, we have considered only two race groups, *clear* and *dark*, suffering problems to find more samples for the *dark* class during our gathering stage. For that reason the training set is smaller than the one used for gender recognition. First row in Figure 2 reflects the results achieved. Due to the unbalanced distribution of the test set, the total success rate is close related to the clear skin class rate, however it is observed that around 30 eigenvalues are necessary to classify correctly the training set, while the best results for the test set are achieved in the range 50 – 70.

**Glasses presence:** For the glasses presence problem, we have restricted the image to the eyes area, see Figure 4.A. Middle row in Figure 2 reflects the results, the test set is correctly classified with around 30 eigenvalues, while the test set starts to lose some performance (observing both sets) with more than 80 eigenvectors.



**Table 1.** Training and test sets. We have tried to build balanced training sets, but for some descriptors one class is not so frequent in our database, and therefore the training set is reduced and the test set has much more samples of the most typical class.

Descriptor	Training set		Test set	
	Female	Male	Female	Male
Gender	1223	1523	835	2246
Descriptor	Clear	Dark	Clear	Dark
Race	574	316	4811	306
Descriptor	No	Yes	No	Yes
Glasses presence	912	692	4042	356
Moustache presence	710	480	4389	426



**Fig. 3.** Moustache conditioned.

**Moustache presence:** For moustache, bottom graphs in Figure 2 presents the results observing that with more than 50 – 60 eigenvalues the test set starts to lose the success rate.

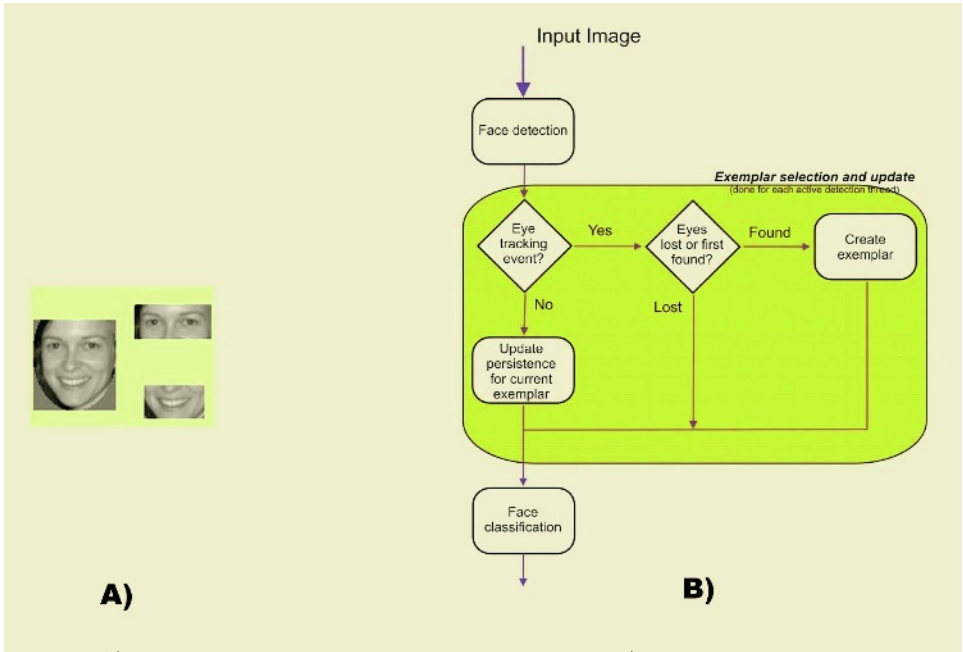
**Conditional classification:** We have also considered the application of a classifier attending to a previous condition. It is supposed that a female has no moustache, therefore, we apply the moustache presence classifier only if the face was considered male. The results reflected in Figure 3 indicates that this information, with the current success rates achieved, improves the performance for the test set.

According to these results, the optimal number of eigenvalues to use are 70 for gender recognition and glasses presence, 60 for moustache presence, and 50 for race classification. In the next section we analyze their performance processing faces automatically detected in video streams.

### 3 Video Stream Processing

Our final objective is to be able to provide the system the ability of describing an individual who interacts with, therefore we apply the conclusions extracted in the previous section to video stream analysis.

For that purpose an automatic face detector is required. The one employed combines the general object detection framework by Viola and Jones [11], skin color detection, tracking and temporal coherence providing high performance, see [1] for more details. For each detected face, the system stores not only its position and size, but also its average color and patterns. In summary,



**Fig. 4.** A) Images areas used for the different problems. B) Exemplar selection process.

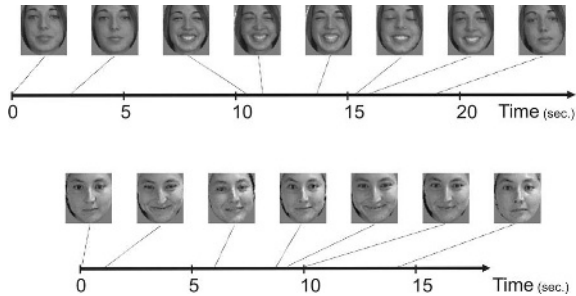
each face detected in a frame can be characterized by different features  $x_i = \langle pos, size, color, eyes_{pos}, eyes_{pattern}, face_{pattern} \rangle$ .

During an interaction session,  $IS$ , the face detector gathers a set of detection threads,  $IS = \{dt_1, dt_2, \dots, dt_n\}$ . A detection thread contains a set of continuous detections, i.e. detections which take place in different frames and are related by the system in terms of position, size and pattern matching techniques. Thus, for each detection thread, the face detector system provides a number of facial samples,  $dt_p = \{x_1, \dots, x_{m_p}\}$ .

### 3.1 Significant Patterns Selection

As mentioned in the previous section, the face detection system provides a set of detection threads. From each, some selected patterns, the exemplars  $e_p = \{e_1, \dots, e_{s_p}\}$ , are extracted in order to reduce information redundancy.

The criteria used to select significant samples in a detection thread, have been chosen to be easily integrated in the detection process. For that reason, it is based on events reported by the the eye tracker integrated in the face detector, see [1] for more details about that detector. A tracking failure shows an evidence of a substantial change in the face appearance, which forces the tracker to lost the target. Under this circumstance, the system needs to use another cue to detect again first the face and later the eyes, or the detection thread will be considered lost. The first face detected in the next frames by the eye tracker is taken as a new exemplar, see Figure 4.B for a graphical overview of the selection



**Fig. 5.** Stable patterns or exemplars extracted from two different detection threads. Dot lines indicates the moment in which they were extracted during interaction.

process. For each exemplar, its time life until the next tracking failure is stored. Therefore, an exemplar is described by the data provided by the normalized detected face,  $x_j$ , and its persistence,  $pe_j$ , i.e.  $e_j = \langle x_j, pe_j \rangle$ . In Figure 5, the exemplars extracted automatically for two individuals during sessions of more than 15 secs. are presented.

Given an interaction session,  $IS$ , for any detection thread,  $dt_p$ , a facial classifier can compute the likelihood for a class,  $C_k$ . This is done by weighting the binary classification for each exemplar according to its relative persistence in relation to the total persistence of the detection thread. This is expressed as:

$$P(C_k|dt_p) = \frac{\sum_{j=1}^{s_p} P(C_k|e_j) * pe_j}{\sum_{n=1}^{s_p} pe_n} \quad (1)$$

### 3.2 Experiments with Video Streams

70 sequences corresponding to different individuals, cameras and environments with a resolution of  $320 \times 240$  were recorded and processed. The total set contains 27271 images, presenting all of them a face easily detected by a human. The face detector located 98.5% of them with an error rate of 5%.

Table 2 summarizes the results for the different descriptors, computed with (1) for the exemplars automatically extracted from each sequence. The correct classification rates are above 80% for moustache and glasses presence problems. For race classification the results are above 90% for both classes, but it must be noticed that the number of dark individuals is reduced in the test set. For gender recognition the results are worse, over 70% for both sets using the eyes area, and over 65% using the whole face.

This low confidence achieved by the gender recognizer can be used by the system to suggest a classification only if the winner class has a likelihood greater than 0.7, asking for supervision in any other situation. This action will additionally allow the system to distinguish who is not correctly classified, and therefore who should be added to the training set, due to the fact that his/her particular data are still not properly considered in the gender model. That information can

**Table 2.** Results achieved for facial description. The left column reflects in brackets the number of individuals (video streams) with a particular feature. The other columns indicate the percentage of those sequences which were labelled with a likelihood of belonging to a class ( $F$  for female,  $\neg F$  for male,  $C$  for clear skin,  $\neg C$  for dark skin,  $G$  for glasses,  $\neg G$  for no glasses,  $M$  for moustache and  $\neg M$  for no moustache). For example, the value in the second row and column, 56.5%, indicates that this percentage of sequences was assigned to the class  $\neg F$ , i.e. Male, with a likelihood greater than 0.7.

	$P(\neg F) > 0.7$	$P(\neg F) > 0.5$	$P(F) > 0.5$	$P(F) > 0.7$
Male (46), using the face	56.5%	65.2%	34.6%	17.3%
Male (46) using the eyes	65.2%	71.7%	28.2%	15.2%
Female (24) using the face	4.1%	4.1%	95.8%	83.3%
Female (24) using the eyes	8.3%	20.8%	79.1%	66.6%
	$P(\neg C) > 0.7$	$P(\neg C) > 0.5$	$P(C) > 0.5$	$P(C) > 0.7$
Clear skin (67)	89.5%	94%	5.6%	1.4%
Dark skin (3)	0%	0%	100%	33.3%
	$P(\neg G) > 0.7$	$P(\neg G) > 0.5$	$P(G) > 0.5$	$P(G) > 0.7$
No glasses (59)	81.3%	86.4%	13.5%	11.8%
With glasses (11)	9%	18%	81%	36%
	$P(\neg M) > 0.7$	$P(\neg M) > 0.5$	$P(M) > 0.5$	$P(M) > 0.7$
No moustache (64)	92.2%	98.4%	1.5%	0%
With moustache (6)	0%	16.6%	83.3%	66.6%

be used by the system to tune the classifier based on its experience, in order to learn iteratively a better classifier.

## 4 Conclusions

An analysis has been performed for facial description in static images and video streams. A subset of the total number of eigenvectors has been empirically selected in order to get better performance for each problem. An approach for significant samples extraction from video streams has also been described. The results achieved classifying automatically selected faces in video streams of individuals not contained in the training set are decent enough to keep on developing these abilities for a machine.

Further work must focus on gathering more interactive sessions with individuals with features less frequent in our test set, to perform further experiments. Additionally, we are interested in developing some tools for self supervision of the system in order to improve the current classifiers by means of its own experience.

## Acknowledgments

Work partially funded by research projects Univ. of Las Palmas de Gran Canaria UNI2003/06, UNI2004/10 and UNI2004/25, Canary Islands Autonomous Government PI2003/160 and PI2003/165 and the Spanish Ministry of Education and Science and FEDER funds (TIN2004-07087).

## References

1. M. Castrillón Santana, J. Lorenzo Navarro, D. Hernández Sosa, and Y. Rodríguez-Domínguez. An analysis of facial description in static images and video streams. In *2nd Iberian Conference on Pattern Recognition and Image Analysis*, Estoril, Portugal, June 2005.
2. O. Déniz Suárez, M. Castrillón Santana, and F. M. Hernández Tejera. Face recognition using independent component analysis and support vector machines. *Pattern Recognition Letters*, 24(13):2153–2157, September 2003.
3. F. Gosselin and P. G. Schyns. Bubbles: a technique to reveal the use of information in recognition tasks. *Vision Research*, pages 2261–2271, 2001.
4. Zhong Jing and Robert Mariani. Glasses detection and extraction by deformable contour. In *International Conference on Pattern Recognition*, 2000.
5. Y. Kirby and L. Sirovich. Application of the karhunen-loève procedure for the characterization of human faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1), July 1990.
6. Christine L. Lisetti and Diane J. Schiano. Automatic facial expression interpretation: Where human-computer interaction, artificial intelligence and cognitive science intersect. *Pragmatics and Cognition (Special Issue on Facial Information Processing: A Multidisciplinary Perspective)*, 8(1):185–235, 2000.
7. Michael J. Lyons, Julien Budyneck, and Shigery Akamatsu. Automatic classification of single facial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1357–1362, December 1999.
8. Baback Moghaddam and Ming-Hsuan Yang. Learning gender with support faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):707–711, 2002.
9. Alex Pentland. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, January 2000.
10. V. Vapnik. *The nature of statistical learning theory*. Springer, New York, 1995.
11. Paul Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, 2001.
12. Bo Wu, Haizhou Ai, and Ran Liu. Glasses detection by boosting simple wavelet features. In *17th Int. Conf. on Pattern Recognition, Cambridge, UK*, pages 292–295, August 2004.

# Recognition of Facial Gestures Based on Support Vector Machines

Attila Fazekas and István Sánta

Faculty of Informatics, University of Debrecen, Hungary  
H-4010 Debrecen P.O.Box 12  
Attila.Fazekas@inf.unideb.hu

**Abstract.** This paper addresses the problem of recognition of emotional facial gestures from static images in thumbnail resolution. More experiments are presented, a holistic and two local approaches using SVM's as classifier engines. The experimental results related to the application of our method are reported<sup>1</sup>.

## 1 Introduction

As the Multimax Principle says, the natural communication is multimodal so far as it can be [1]. Thus our aim must be to create similar multimodalities in the communicative way of human-computer interaction. Faces are our interfaces in our emotional and social lives. They should take part in our communication with computers as well. The face is our unique feature. Even the faces of the twins differ in some respects. Humans can detect the differences between two faces very easily, but this is a hard task for a computer.

Automatic analysis of facial gestures is rapidly becoming an area of intense interest in multi-modal human-computer interaction. However, the basic goal of this area of research mapping detected facial gestures into a human-like description of shown facial expression is yet to be achieved.

Pantic and Rothkrantz [12] have laid down the basic requirements of an ideal system for facial expression analysis. The first of these points is the requirement of fully automatic system, i.e. it has to automatically perform all the stages of the recognition (face detection, feature extraction and expression classification). There have already been some experiments for developing such a system e.g. [13].

The published studies cover almost all the possible approaches for facial expression recognition [4, 7, 12]. There are holistic [2] and local image-based [11] recognizers. Researchers use templates [7], principal components or Gabor filters [2], neural networks [11] for finding the proper expressions. They try to classify the images into facial action units [3], or some main emotion categories [14] from static images [13] or image sequences [3].

Our aim is to recognize the main emotional expressions from static images. In our previous work we provided a holistic approach using Support Vector Machines for classifying sadness, surprise, anger, happiness and neutral expression

---

<sup>1</sup> Research supported by OTKA grants F043090.

from thumbnail images, in which only the main facial regions appear (without hair information) [6]. The thumbnail images give minimal amount of face information to the recognition system. In the present study, we demonstrate two local approaches near a holistic one by using only some local areas of the face, which are important information sources from the aspect of classifying facial gestures. We are using thumbnail resolution and SVM classifier again.

## 2 Support Vector Machine

Statistical learning from examples aims at selecting from a given set of functions  $\{f_\alpha(\mathbf{x}) \mid \alpha \in \Lambda\}$ , the one which predicts best the correct response (i.e. the response of a supervisor). This selection is based on the observation of  $l$  pairs that build the training set:

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l), \quad (1)$$

which contains input vectors  $\mathbf{x}_i \in \mathbb{R}^m$  and the associated ground “truth”  $y_i \in \{+1, -1\}$  given by an external supervisor.

Let the response of the learning machine  $f_\alpha(\mathbf{x})$  belongs to a set of indicator functions (which admits the value  $+1$  or  $-1$ )  $\{f_\alpha(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^m, \alpha \in \Lambda\}$ . If we define the loss-function:

$$L(y, f_\alpha(\mathbf{x})) = \begin{cases} 0, & \text{if } y = f_\alpha(\mathbf{x}), \\ 1, & \text{if } y \neq f_\alpha(\mathbf{x}) \end{cases} \quad (2)$$

that measures the error between the ground truth  $y$  to a given input  $\mathbf{x}$  and the response  $f_\alpha(\mathbf{x})$  provided by the learning machine, the expected value of the loss is given by:

$$R(\alpha) = \int L(y, f_\alpha(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy \quad (3)$$

where  $p(\mathbf{x}, y)$  is the joint probability density function of random variables  $\mathbf{x}$  and  $y$ .  $R(\alpha)$  is called the expected risk. We would like to find the function  $f_{\alpha_0}(\mathbf{x})$  which minimizes the risk functional  $R(\alpha)$ . The selection of the function is based on the training set of  $l$  random independent identically distributed observations (1) [8].

The basic idea of SVM to construct the optimal separating hyperplane. Suppose that the training data (1) can be separated by a hyperplane,  $f_\alpha(\mathbf{x}) = \alpha^T \mathbf{x} + b = 0$ , such that:

$$y_i (\alpha^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, l \quad (4)$$

where  $\alpha$  is the normal to the hyperplane,  $\frac{|b|}{\|\alpha\|}$  is the perpendicular distance from the hyperplane to the origin, and  $\|\alpha\|$  is the Euclidean norm of  $\alpha$ . Let  $d_+$  ( $d_-$ ) be the Euclidean distance from the separating hyperplane to the closest positive (negative) example. The margin of the separating hyperplane is defined to be

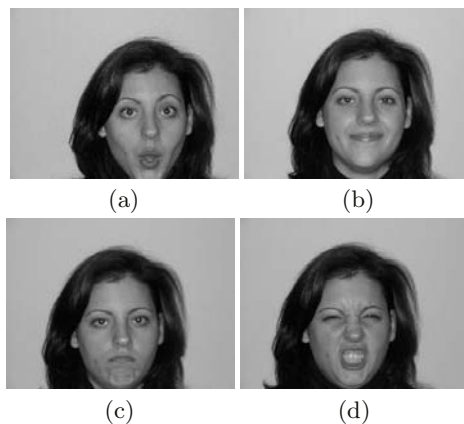
$d_+ + d_-$ . For the linearly separable case, SVM simply seeks for the separating hyperplane with the largest margin [10, 15–17].

For linearly non-separable data, by mapping the input pattern vectors, which are the elements of the training set, into a high-dimensional feature space through an a priori suitably chosen mapping, we expect that the elements of the training set will be linearly separable in the feature space. We construct the optimal separating hyperplane in the feature space to get a binary decision whether the input vector belongs to a given class or not. For example, in the case of the application studied in the paper, facial gesture recognition, the input vector comprises gray levels of pixels from a rectangular region of the digital image and the result of the binary decision is the answer whether this region, for example, is a smiling face or not.

In this research, we used the application *SVM*<sup>Light</sup> developed by T. Joachims [9].

### 3 Image Database

In the presented study we used our face database of 600 images. All of its images are recorded in 256 gray levels and are of dimension  $640 \times 480$ . These images show the head of 40 different subjects asked for performing 5 facial gestures (neutral, sad, surprised, angry, and smiling) successively. We repeated this sequence three times. Figure 1 shows surprising, smiling, sad, and angry face in the original resolution.



**Fig. 1.** Surprising face (a), smiling face (b), sad face (c), angry face (d) in the original resolution.

Our dataset is divided into three disjoint parts. All the images of 6 randomly selected persons are referred as database *TEST\_ONLY*. This is the collection of images of subjects, who are absolutely unknown for the SVM's, i.e. who are not trained to them only tested. The gesture sequence recorded at 3rd time of the remaining 34 subjects makes up the database called *TEST*. This is the collection



of images, which are unknown for the SVM's of persons, whose other 2 sequences belong to the database TRAIN and used for training the classifiers.

In our preliminary works we tried to classify the gestures from the image of the whole face. From each image, a bounding rectangle of dimension  $256 \times 320$  pixels has been manually determined that includes the actual face. This area has been subsampled four times in the following way: at each subsampling, non-overlapping regions of  $2 \times 2$  pixels are replaced by their average. [5] The size of this dataset is multiplied by 5 by shifting each  $16 \times 20$ -pixel image of database TRAIN by a pixel along the four main direction for decreasing the error of the manual recording of rectangles.

In our succeeding studies, we tried a local approach by determining the bounding rectangle of the eyes and their neighbourhood, and that of the mouth and its surroundings. The eye-rectangles have dimension  $208 \times 96$  pixels and the mouth-rectangles have dimension  $184 \times 96$  pixels. These patterns have been subsampled three times according to the above procedure. Figure 2 shows smiling eyes and mouth.



**Fig. 2.** Smiling eyes (a), smiling mouth (b) in the original resolution.

## 4 Experimental Results

In all the presented experiments, the groundtruth of each gesture contains all the vectors of the database TRAIN labeled with +1, if the given pattern shows the given gesture, and with -1, otherwise.

In learning phases, a lot of different kernel functions were tried. At all times, we found that different gestures can be recognized on the best performance level by using different kernels.

After the learning phases, the trained SVM's are tested on database TEST and TEST\_ONLY. We measured the classification errors, i.e. the sum of false positive and false negative answers, SVM by SVM.

In the holistic case, we trained 5 SVM's, one for each gesture. We used linear, and 2nd, 3rd and 4th degree polynomial kernels and tested all of them on the test sets. Table 1 shows the best results. The classification errors are in terms of percent.

In the local approach, we trained two individual SVM's for each gesture, one for the eyes, the other for the mouth, with the proper groundtruth. We measured the classification errors of these SVM's for different linear and polynomial kernels and the 3rd degree kernels seemed to produce the best performance. We selected the best one of them for the additional experiments. Its results can be seen in Table 2 for the eyes and table 3 for the mouths.

**Table 1.** Experimental results of the holistic approach.

	TEST	TEST_ONLY
Neutral linear kernel	21.76%	32.22%
Sad 3rd degree polynomial kernel	22.35%	30.00%
Surprised 3rd degree polynomial kernel	11.76%	24.44%
Angry 3rd degree polynomial kernel	14.12%	15.56%
Happy linear kernel	14.12%	14.44%

**Table 2.** Experimental results of the local approach. Results of the eyes-SVM's.

	TEST	TEST_ONLY
Neutral	15.88%	30.00%
Sad	21.18%	32.22%
Surprised	14.12%	15.56%
Angry	15.88%	16.67%
Happy	18.82%	22.22%

**Table 3.** Experimental results of the local approach. Results of the mouth-SVM's.

	TEST	TEST_ONLY
Neutral	21.76%	27.78%
Sad	17.06%	28.89%
Surprised	17.65%	22.22%
Angry	18.82%	22.22%
Happy	7.65%	3.33%

As can be seen, these results have greater scatter, because there can be different gestures with very similar eye-state or mouth-state.

In what follows, the outputs of these 10 SVM's with the selected kernels for each eyes-mouth pair have made a new groundtruth of dimension 10. In this second layer, there are 5 SVM's trained for each gesture. Again, each SVM receives the 10-dimension vector of all the pairs labeled with +1, if the pair belongs to the given gesture, and with -1, otherwise.

For reducing the classification errors of the first layer, we used the outputs of the misclassified images of the training set as well as the right outputs (of course with the proper labels) for training the second layer.

For the SVM's of the second layer, we tried some polynomial kernels. The best performance is showed by Table 4.

**Table 4.** Experimental results. Results of the 2-layer SVM network.

	TEST	TEST_ONLY
Neutral 3rd degree polynomial kernel	12.94%	14.44%
Sad 2nd degree polynomial kernel	14.12%	22.22%
Surprised 2nd degree polynomial kernel	12.94%	15.56%
Angry 3rd degree polynomial kernel	15.29%	14.44%
Happy 3rd degree polynomial kernel	5.88%	4.44%

As we expected, the happiness is significantly more recognizable than other 4 gestures, which are on approximately the same significance level.

## 5 Conclusions

We have presented some progressing experiments for classifying emotional facial gestures using still images in thumbnail resolution. We have used SVM classifiers. First, a holistic approach has been accomplished using thumbnail images of the whole face. Its test results have proved to be acceptable. However, it can be improved using local procedures as it can be seen in results of our further studies. They have used two local parts of the faces, eyes and mouth with their surroundings. Our latest reported experiment using two-layer SVM network have turned out to be a much stronger classifier than our previous not-so-bad procedures.

Since the direct transition between certain states of the face may happen rather rarely, thus information retrieved from more than one immediately succeeding frames of video sequences can increase on the accuracy. It will be the next step in our research work.

## References

1. Bunt, H.: Issues in Multimodal Human-Computer Communication. Lecture Notes in Computer Science **1374** (1998) 1–12
2. Dailey, M.N., and Cottrell, G.W.: PCA = Gabor for Expression Recognition. Technical Report CS-629, UCSD (1999)
3. Donato, G., Bartlett, M.S., Hager, J.C., Ekman, P., and Sejnowski, T.J.: Classifying Facial Actions. IEEE Transactions on Pattern Analysis and Machine Intelligence **21**(10) (1999) 974–989
4. Fasel, B., and Luetttin, J.: Automatic Facial Expression Analysis: A Survey. Pattern Recognition **36**(1) (2003) 259–275
5. Fazekas, A., Kotropulos, C., Buciu, I., and Pitás, I.: Support vector machines on the space of Walsh functions and their properties. In Proc. of 2nd International Symposium on Image and Signal Processing and Analysis (19-21 June, 2001, Pula, Croatia) 43–48

6. Fazekas, A., and Sánta, I.: Recognition of Facial Gestures From Thumbnail Picture. In Proc. of NOBIM'2004 (27-28 May, 2004, Stavanger, Norway) 54–57
7. Fellenz, W.A., Taylor, J.G., Tsapatsoulis, N., and Kollias, S.: Comparing template-based, feature-based and supervised classification of facial expressions from static images. In Proceedings of Circuits, Systems, Communications and Computers (CSCC '99) (1999) 5331–5336
8. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, San Diego (1990)
9. Joachims, T.: Making large-scale SVM learning practical. in Advances in Kernel Methods - Support Vector Learning, MIT Press, Cambridge, MA (1999) 169–184
10. Osuna, E.E., Freund, R., and Girosi, F.: Support vector machines: Training and applications. CBCL Technical Report (March 1997) 1–41
11. Padgett, C., Cottrell, G.W., and Adolphs, R.: Categorical perception in facial emotion classification. In Proceedings of The Eighteenth Annual Conference of the Cognitive Science Society, San Diego, CA (1996) 249–253
12. Pantic, M., and Rothkrantz, L.J.M.: Automatic Analysis of Facial Expressions: The State of the Art. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(12) (December 2000) 1424–1444
13. Pantic, M., Rothkrantz, L.J.M.: Expert system for automatic analysis of facial expressions. Image and Vision Computing **18** (2000) 881–905
14. Silapachote P., Karuppiyah, D.R., and Hanson, A.R.: Feature selection using AdaBoost for face expression recognition. In Proceedings of The 4th IASTED International Conference on Visualization, Imaging, and Image Processing, VIIP 2004, Marbella, Spain (September 2004) 84–89
15. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)
16. Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
17. Vapnik, V.N.: An overview of statistical learning theory. IEEE Transactions on Neural Networks **10**(5) (September 1999) 988–999

# Performance Driven Facial Animation by Appearance Based Tracking

José Miguel Buenaposada<sup>1</sup>, Enrique Muñoz<sup>2</sup>, and Luis Baumela<sup>2</sup>

<sup>1</sup> Dpto. de Informática, Estadística y Telemática  
ESCET, Univ. Rey Juan Carlos  
C/ Tulipán, s/n, 28933 Móstoles, Madrid, Spain  
`jmbuenaposada@escet.urjc.es`

<sup>2</sup> Fac. de Informática, Univ. Politécnica de Madrid  
Campus de Montegancedo s/n, 28660 Boadilla del Monte, Madrid, Spain  
`kike@dia.fi.upm.es`, `lbaumela@fi.upm.es`

**Abstract.** We present a method that estimates high level animation parameters (muscle contractions, eye movements, eye lids opening, jaw motion and lips contractions) from a marker-less face image sequence. We use an efficient appearance-based tracker to stabilise images of upper (eyes and eyebrows) and lower (mouth) face. By using a set of stabilised images with known animation parameters, we can learn a re-animation matrix that allows us to estimate the parameters of a new image. The system is able to re-animate a 32 DOF 3D face model in real-time.

## 1 Introduction

Automated computer animation of faces and avatars is an area of intense research for its application in the television, computer games and film industry. Performance driven animation is usually done by motion capture using markers on the face. Computer vision provides an alternative non-intrusive marker-less approach to motion capture.

Generally, the face shapes of the actor and that of the animated model are different. So, a method to adapt the motion of the former to the latter is needed [1]. There are two ways to achieve this: parametrisation and motion modification. By facial motion modification we mean to adapt the vertex deformation due to facial motion to the new facial model. In [1] were introduced some algorithms and heuristics to translate the facial expression motion from a facial model into another with different surface structure. Procedures based on parametrisation aim to describe motion with a set of values that, when applied to any facial model, will produce a similar expression. Among the parametrised systems we can distinguish those that use standard facial expressions coding, like FACS[2, 3] or MPEG-4 FAPS [4, 5], and those that use and *ad-hoc* coding [6, 7]. When the abstraction level of the animation parameters is high, then the estimation of these parameters is more difficult. This is due mainly to the weak relationship between image measurements and control parameters.

In this paper we present a method that estimates high level animation parameters from a marker-less face image sequence. We will use a muscle-based 3D face model resulting in a parametrised motion capture algorithm. We have previously developed an efficient appearance based tracker [8] that locates and tracks the eyes and the mouth in spite of the non-rigid motion of the face. The main contribution of this paper is a procedure to estimate the animation parameters of a 3D face model from stabilised images of the eyes and mouth obtained from our tracking algorithm. This procedure is composed of two training steps, one for building an eigenspace for tracking, and another one for learning a linear relation between the animation parameters and the stabilised images. In the following sections we will present this algorithm and some results.

## 2 Appearance Based Tracking

The tracking algorithm presented in this section can be seen as an extension of the Hager and Belhumeur's *Jacobian factorisation* [9] where we impose no restrictions on the PCA-based subspace model used. It is also related to the Black and Jepson's *Eigenttracking* [10], but instead of computing the motion parameters by using a gradient descent procedure in which the target image Jacobian must be computed for each frame in the sequence, as in [10], we use a set of precomputed motion templates which alleviate the computations that have to be performed on line.

Let  $P$  be the image of a target. The subspace constancy equation holds for all pixels in the target [10]:

$$I(f(\mathbf{x}, \boldsymbol{\mu}), t) = [\mathbf{Bc}(t)](\mathbf{x}) \quad \forall x \in P, \quad (1)$$

where  $\mathbf{x}$  is the vector of co-ordinates of a point in image  $I$ ,  $\mathbf{B}$  is the subspace base matrix,  $\mathbf{c}$  is the vector of subspace coefficients, and  $I(f(\mathbf{x}, \boldsymbol{\mu}), t)$  is the image acquired at time  $t$  rectified with motion model  $f(\mathbf{x}, \boldsymbol{\mu})$  and motion parameters  $\boldsymbol{\mu}$ . By  $[\mathbf{Bc}](x)$  we denote the value of  $\mathbf{Bc}$  for the pixel with position  $\mathbf{x}$  in the image. Matrix  $\mathbf{B}$  is of dimension  $N \times k$ , where  $N$  is the number of pixels per image and  $k$  is the number of basis vectors in the subspace. Intuitively (1) states that the rigidly rectified image  $I(f(\mathbf{x}, \boldsymbol{\mu}), t)$  can be expressed as a linear combination of the appearance subspace basis vectors,  $\mathbf{B}$ <sup>1</sup>.

Tracking consists on estimating for each image in the sequence the values of the motion,  $\boldsymbol{\mu}$ , and appearance,  $\mathbf{c}$ , parameters which minimise the error function

$$E(\boldsymbol{\mu}, \mathbf{c}) = \|\mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}_t), t) - \mathbf{Bc}(t)\|^2,$$

where  $\mathbf{I}(\mathbf{x})$  is  $I(\mathbf{x})$  in vector form (scanning  $I$  by rows or columns). In order to make Gauss-Newton iterations, a Taylor series expansion of  $\mathbf{I}$  at  $(\mathbf{x}, t)$  is performed, producing a new error function

$$E(\delta\boldsymbol{\mu}, \mathbf{c}) = \|\mathbf{M}\delta\boldsymbol{\mu} + \mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu})) - \mathbf{Bc}\|^2,$$

<sup>1</sup> We assume that that the average image has been included as the first column of  $\mathbf{B}$ .

where  $\mathbf{M} = \frac{\partial \mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}))}{\partial \boldsymbol{\mu}}$  is the  $N \times n$  ( $n = \dim(\boldsymbol{\mu})$ ) Jacobian matrix of  $I$  (note that dependence on  $t$  has been dropped for convenience). In the following subsections we will introduce a procedure for precomputing a set of motion templates which efficiently minimise (2) for any linear subspace model.

## 2.1 Jacobian Matrix Factorisation

One of the obstacles for minimising (2) on line, while tracking, is the computational cost of estimating  $\mathbf{M}$  for each frame. Following an approach similar to [9],  $\mathbf{M}$  can be expressed in terms of the gradient of the subspace basis vectors,  $\mathbf{B}_{\nabla}$ , which are constant, and the motion and appearance parameters  $(\boldsymbol{\mu}, \mathbf{c})$ , which vary over time. If we choose a motion model  $f$  such that  $\mathbf{C}f_{\mathbf{x}}(\mathbf{x}_i, \boldsymbol{\mu})^{-1}f_{\boldsymbol{\mu}}(\mathbf{x}_i, \boldsymbol{\mu}) = \Gamma(\mathbf{x}_i)\boldsymbol{\Sigma}(\boldsymbol{\mu}, \mathbf{c})$ , then  $\mathbf{M}$  can be factored into

$$\mathbf{M}(\boldsymbol{\mu}, \mathbf{c}) = \begin{bmatrix} \mathbf{B}_{\nabla}(\mathbf{x}_1)\Gamma(\mathbf{x}_1) \\ \vdots \\ \mathbf{B}_{\nabla}(\mathbf{x}_N)\Gamma(\mathbf{x}_N) \end{bmatrix} \boldsymbol{\Sigma}(\boldsymbol{\mu}, \mathbf{c}) = \mathbf{M}_0\boldsymbol{\Sigma}(\boldsymbol{\mu}, \mathbf{c}),$$

where  $\mathbf{B}_{\nabla}(\mathbf{x}_i)$  is the Jacobian of  $\mathbf{B}$  with respect to the image co-ordinates. Then  $\mathbf{M}_0$  is a constant matrix and  $\boldsymbol{\Sigma}$  depends on  $\mathbf{c}$  and  $\boldsymbol{\mu}$ .

## 2.2 Minimising $E(\boldsymbol{\mu}, \mathbf{c})$

As  $\mathbf{M}$  depends on both,  $\boldsymbol{\mu}$  and  $\mathbf{c}$ , (2) defines a nonlinear cost function over  $\delta\boldsymbol{\mu}$  and  $\mathbf{c}$ . The optimisation algorithm that we use first assumes  $\mathbf{c}$  constant and computes the minimum of  $E(\boldsymbol{\mu}, \mathbf{c})$  w.r.t.  $\boldsymbol{\mu}$ ,

$$\delta\boldsymbol{\mu} = -(\boldsymbol{\Sigma}^{\top}\mathcal{M}\boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}^{\top}\mathbf{M}_0^{\top}[\mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}), t + \tau) - \mathbf{B}\mathbf{c}(t)],$$

where  $\mathcal{M} = \mathbf{M}_0^{\top}\mathbf{M}_0$ . Then it minimises  $E$  over  $\mathbf{c}$  assuming  $\boldsymbol{\mu}$  constant,

$$\mathbf{c} = \mathbf{B}^{\top}[\mathbf{M}\delta\boldsymbol{\mu} + \mathbf{I}(f(\mathbf{x}, \boldsymbol{\mu}), t + \tau)].$$

Once we have  $\mathbf{c}$ , we can refine the estimation of  $\delta\boldsymbol{\mu}$  by using (2.2) again. Normally two or three iterations are enough to reach a stable solution. We have developed the factorisation for the rotation-translation-scale, the affine and the projective motion models [8]. In this paper we will use a projective motion model,  $f(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{H}\mathbf{x}$ , where  $\mathbf{H}$  is a  $3 \times 3$  homography.

## 3 Reanimation

The philosophy to performance driven animation of a 3D face model we propose, is similar to the Valente and Dugelay's one [4]. We will use stabilised view images of the user's eyes and mouth with known animation parameters to estimate a linear relationship between grey levels and animation parameters. In order to

estimate the control parameters of their face model, Valente and Dugelay use optical flow and not raw grey levels as we do. They use a very realistic 3D face model of each particular user. Therefore, by driving their model with a set of control parameters it was possible to get the corresponding optical flow for each face region. Valente and Dugelay use a feature based tracker (five features) and a Kalman filter to get the normalised images of different face regions. As their tracker is not designed to deal with non-rigid motion, it is not clear how is it going to work with extreme facial expressions.

In our case, the appearance based tracker of section 2 allows us to track the most informative face areas in spite of the non-rigid motion due to facial expressions. With the tracker we can extract stabilised images of any part of the face for each frame in the sequence. In this section we are going to show how to estimate the face animation parameters from stabilised images of the lower and the upper part of the face.

### 3.1 Animation Parameters Estimation

In order to estimate the animation parameters for a given face region we will use  $e$  example images each with  $N$  pixels. Let  $\mathbf{I}$  be an  $N \times e$  matrix, where each column  $\mathbf{i}_j$  has one of the example images (e.g. scanning the image by rows), and let  $\mathbf{A}$  be an  $a \times e$  matrix, where each column  $\mathbf{a}_j$  represents the animation parameters,  $\mathbf{a}$ , corresponding to the appearance in  $\mathbf{i}_j$ <sup>2</sup>. Then  $\mathbf{D}_e$  is an  $(N + a) \times e$  matrix:

$$\mathbf{D}_e = \begin{bmatrix} \mathbf{I} \\ \mathbf{W}_A \mathbf{A} \end{bmatrix} = \begin{bmatrix} \mathbf{i}_1 & \cdots & \mathbf{i}_e \\ \mathbf{W}_A [\mathbf{a}_1 & \cdots & \mathbf{a}_e] \end{bmatrix}, \quad (2)$$

where  $\mathbf{W}_A$  is a diagonal matrix of weights that takes into account the different scale of the animation parameters and grey levels. The weight matrix we use, is  $r\mathbf{I}$  where  $r^2$  is the rate between the grey levels variability and total variability in the animation parameters. In the Direct Appearance Models framework [11] it is used a similar matrix but for grey levels and shape parameters.

By computing PCA of matrix  $\mathbf{D}_e$ , we get  $\mathbf{B}_l$ , the subspace basis expanded by the  $l$  eigenvectors<sup>3</sup> corresponding to the bigger eigenvalues of the covariance matrix ( $\mathbf{D}_e \mathbf{D}_e^T$ ), which can be written as

$$\mathbf{B}_l = \begin{bmatrix} \mathbf{B}_i \\ \mathbf{B}_a \end{bmatrix}.$$

Using the  $(N + a) \times l$  matrix,  $\mathbf{B}_l$ , the vector  $\mathbf{c}_l$ , that represents the relation between the images in  $\mathbf{I}$  and the animation parameters in  $\mathbf{A}$  can be estimated. By using  $\mathbf{c}_l$ , we can approximate each pair  $(\mathbf{i}, \mathbf{a})$  by  $(\mathbf{i}^*, \mathbf{a}^*)$  in such a way that:

$$\begin{bmatrix} \mathbf{i}^* \\ \mathbf{W}_A \mathbf{a}^* \end{bmatrix} = \mathbf{B}_l \mathbf{c}_l, \mathbf{c}_l = \mathbf{B}_l^T \begin{bmatrix} \mathbf{i} \\ \mathbf{W}_A \mathbf{a} \end{bmatrix}.$$

<sup>2</sup> We assume, that all examples,  $\mathbf{i}_j$ , and animation parameters,  $\mathbf{a}_j$ , are mean centred.

<sup>3</sup> Note that we use two eigenspaces, one for tracking and the other for reanimation.



Given an image  $\mathbf{i}$ , and  $\mathbf{B}_i$  and  $\mathbf{B}_a$  matrices from training, the re-animation problem is to estimate the corresponding animation parameters,  $\mathbf{a}^*$ . From the structure of  $\mathbf{B}_l$  we can write  $\mathbf{B}_i \mathbf{c}_l = \mathbf{i}$ , where  $\mathbf{c}_l$  is the only unknown. In general, the number of image pixels  $N$  is much bigger than  $l$  and the solution for  $\mathbf{c}_l$  will be given by the minimisation of

$$\mathbf{c}_l^* = \arg \min_{\mathbf{c}_l} \|\mathbf{B}_i \mathbf{c}_l - \mathbf{i}\|^2 = \text{pinv}(\mathbf{B}_i) \mathbf{i}, \quad (3)$$

where the  $l \times N$  matrix  $\text{pinv}(\mathbf{B}_i)$ , is the pseudo-inverse of  $\mathbf{B}_i$  computed by using SVD. And then, the animation parameters that corresponds to the image  $\mathbf{i}$  are given by

$$\mathbf{W}_A \mathbf{a}^* = \mathbf{B}_a \text{pinv}(\mathbf{B}_i) \mathbf{i} = \mathbf{R}_i^a \mathbf{i}, \quad (4)$$

where the  $a \times N$  matrix,  $\mathbf{R}_i^a$ , is constant and can be precomputed. As we get  $\mathbf{W}_A \mathbf{a}^*$  from (4), it is needed to multiply it by  $(\mathbf{W}_A)^{-1}$  in order to obtain the animation parameters estimation,  $\mathbf{a}^*$ , in the right scale.

## 4 Experiments

In all the experiments conducted<sup>4</sup> in this section the face is split in the upper face (the eyes region) and the lower face (mouth region) areas. As the motion of the two regions is almost independent we can build two appearance models needing less examples on each (a modular eigenspace). Nevertheless, our tracker uses the grey levels from both regions to compute motion parameters but maintaining separate appearance parameters.

### 4.1 Quantitative Experiments

In the first experiment we would like to assert the quality of the re-animation. To do so, we use a modified version of the Parke and Waters' 3D face model [12] with 32 degrees of freedom. The 3D face model is used to render three image sequences: a training sequence for the eyes (630 images), a training sequence for the mouth (540 images) and a test sequence (1225 images, see figure 1). The facial expressions in the test sequence are different from the ones used in the training sequences.



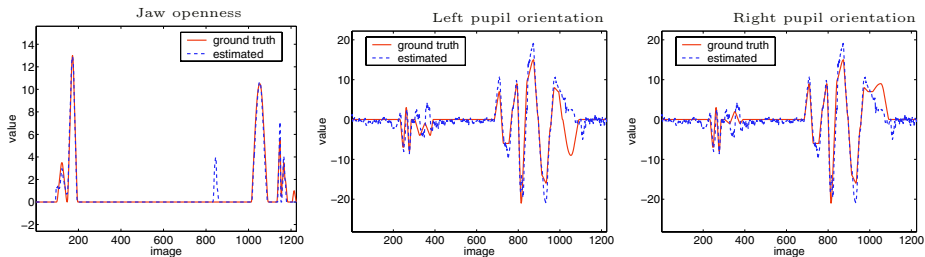
**Fig. 1.** Some of the 75 key-frames used to render the test sequence (1125 images).

<sup>4</sup> See videos in <http://www.dia.fi.upm.es/~lbaumela/FaceExpressionRecognition/>

In the eyes training sequence there is only non-rigid motion in the upper area of the face. Therefore, the stabilised images of the eyes can be extracted automatically by tracking the mouth area with a simple template tracker (using a mouth template). Similarly, as in the mouth training sequence there is only non-rigid motion in the lower face, the mouth stabilised images are computed by rigidly tracking the eyes. We have extracted a region of the eyes with  $N_{eyes} = 60 \times 35$  pixels and a region of the mouth with  $N_{mouth} = 53 \times 43$  pixels (that will be used both in tracking and re-animation). The normalised images of the 3D model (from the two training sequences) and the ground truth animation parameters allows us to compute  $R_a^i$ , for each of the face regions (upper and lower face).

In the experiment conducted we use the projective motion model for appearance based tracking. In order to compute the eigenspace matrix for tracking,  $B$ , we use all the training normalised images. For computing the re-animation matrix,  $R_a^i$ , we use the 540 and 629 example pairs (images and animation parameters) for eyes and mouth, respectively.

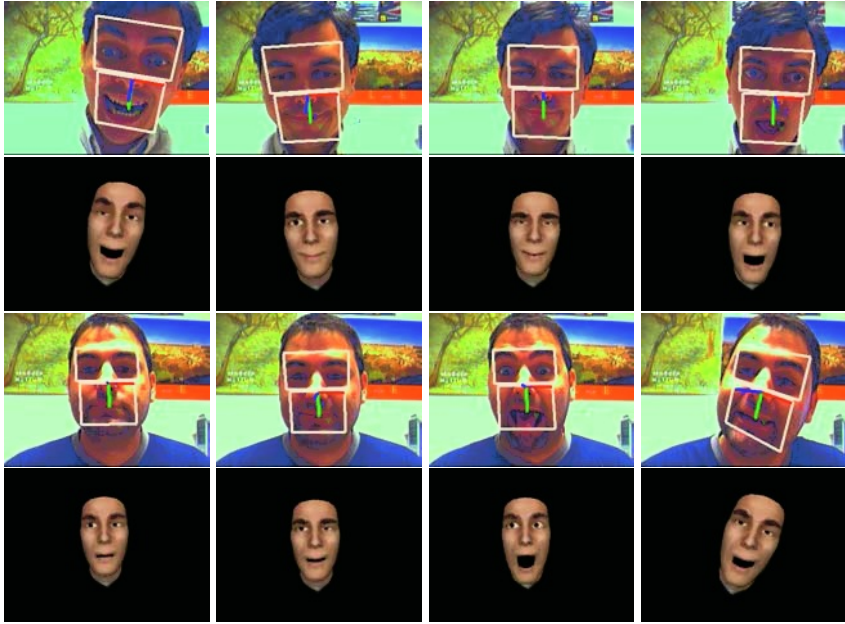
The jaw opening parameter (see figure 2 left) is estimated very accurately except around the frame 830 in which the face is out of the frontal position to the camera. The overall estimation of the pupil horizontal orientation (see figure 2 middle and right) is quite good except in frames 222 to 435, in which the face is not frontal to the camera, and around frame 1050, in which the model is cross-eyed (and we don't have such configuration in the examples).



**Fig. 2.** Synthetic experiment results. 8 On the left, it is shown the estimated jaw openness, in the middle the estimated horizontal rotation for the left pupil and on the right the horizontal orientation of the right pupil.

## 4.2 Qualitative Validation

We have tested our re-animation system with five different users. The main problem here is the selection of the examples for re-animation (the pairs normalised images, animation parameters). The solution we have adopted is to use a set of known face expressions in the 3D model (key frames) and select manually the corresponding normalised images of the user's eyes (21 examples) and mouth (18 examples). By doing so, we get the set of examples needed for the re-animation training. We use all the training normalised images for computing the eigenspace tracking matrices,  $B$ .



**Fig. 3.** Results of two of the qualitative experiments. In first row, appearance based tracking results for first user (the two face areas locations are overlaid in white) and in second row animation results. In third row, appearance based tracking results for the second experiment and fourth row animation results.

All the qualitative experiments were made by taking a very long sequence of images and using half of the sequence for training and the other half for tracking. In figure 3 are shown some of the results for two of the experiments. In the first experiment we used a 4925 image sequence: 2190 images for training and 2735 for testing. And in the second one we used a 4421 images sequence: 2360 images for training and 2061 images for testing. Due to lack of space we can not show all the five re-animation experiments.

## 5 Conclusions

In this paper we have shown one of the applications of facial analysis: performance driven animation. The animation system presented can be adapted, by training, to any user and illumination conditions and the current implementation of our appearance based tracker (not optimised) can track the upper part of the face at 25 fps and the whole face at 15 fps. Given that the re-animation only needs the multiplication of matrix  $R_i^a$  by the grey levels of the corresponding normalised image, it allows the animation of the 3D model in real time.

Some issues still remain open. The adaptation to a new user is in part manual, mainly because we have not studied how to choose automatically the user images

that correspond to facial expressions in the 3D model. We are currently building a robust tracker, which efficiently deal with occlusions and gross illumination changes.

## Acknowledgement

The authors gratefully acknowledge funding from the Spanish Ministry of Science and Technology under grant number TIC2002-000591. Enrique Muñoz was funded by a FPU grant from the Spanish Ministry of Education. We also thanks the anonymous reviewers of this paper for their feedback.

## References

1. Jun-yong, Neumann, U.: Expression cloning. In: Proc. of SIGGRAPH, ACM (2001) 277–288
2. Cohn, J., Kanade, T., Moriyama, T., Ambadar, Z., Xiao, J., Ga, J., Imamura, H.: A comparative study of alternative face coding algorithms. Technical report, Robotics Institute, Carnegie Mellon University (2001)
3. Tian, Y., Kanade, T., Cohn, J.: Recognizing action units for facial expression analysis. PAMI **23** (2001) 97–115
4. Stéphane Valente, Ana C. Andrés Del Valle, J.L.D.: Analysis and reproduction of facial expressions for realistic communicating clones. Journal of VLSI Signal Processing **29** (2001) 41–49
5. Ahlberg, J.: Using the active appearance algorithm for face and facial feature tracking. In: Proc. of 2nd Int. Workshop on Recognition, analysis and tracking of faces and gestures in real time systems, RATFG-RTS'01. (2001) 68–72
6. Terzopoulos, D., Waters, K.: Analysis and synthesis of facial image sequences using physical and anatomical models. PAMI **15** (1997)
7. Buck, I., Finkelstein, A., Jacobs, C., Klein, A., Salesin, D.H., Seims, J., Szeliski, R., Toyama, K.: Performance-driven hand-drawn animation. In: Proc. of Int. Symposium on Non Photorealistic Animation and Rendering, NPAR'2000. (2000) 101–108
8. Buenaposada, J., Muñoz, E., Baumela, L.: Efficient appearance-based tracking. In: Proc. of Workshop on Nonrigid and Articulated Motion, IEEE (2004)
9. Hager, G., Belhumeur, P.: Efficient region tracking with parametric models of geometry and illumination. PAMI **20** (1998) 1025–1039
10. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. IJCV **26** (1998) 63–84
11. Hou, X., Li, S.Z., Zhang, H., Cheng, Q.: Direct appearance models. In: Proc. of CVPR, IEEE (2001)
12. Parke, F.I., Waters, K.: Computer Facial Animation. AK Peters Ltd (1996)

# Color Distribution Tracking for Facial Analysis

Juan José Gracia-Roche, Carlos Orrite, Emiliano Bernués, and José Elías Herrero

Computer Vision Lab, Aragon Institute for Engineering Research  
University of Zaragoza, María de Luna 1, 50018 Zaragoza, Spain  
{jjroche, corrite, ebr, jelias}@unizar.es

**Abstract.** In this paper we address the problem of real time object tracking in complex scenes under dynamically changing lighting conditions. This problem affects video-surveillance applications where object location must be known at any time. We are interested in locating and tracking people in video sequences for access control and advanced user interface applications. Here we present a real time tracking method suitable for human faces. A Skin Probability Image (SPI) is generated by applying a skin hue model to the input frame. Targets are located by applying a modified mean-shift algorithm. To obtain their spatial extent, error ellipses are fitted to the probability distributions representing them. The hue model is unique for each target and it is updated each frame to cope with lighting variations. This technique has been applied to human face tracking in indoor environments to test its performance in different situations.

## 1 Introduction

One of the main research lines in our workgroup is facial analysis and recognition. Face detection and tracking are two steps required for the development of applications such as access control or advanced user interfaces. In this paper we introduce a novel approximation to face tracking based on the tracking of color probability distributions, a technique that has been widely used in previous works [1-6].

Color is used in computer vision as an efficient technique to segment and classify image areas. It has important advantages such as pose, occlusions and size invariance. For skin color, it is even possible to find a model common to all people.

In the detection step, color is used to generate a bi-dimensional target probability map according to a statistical model in different color spaces [9]. Skin tone allows different modeling alternatives. In [1, 2] a gaussian model in the RG normalized chromatic space is used and in [3] as a gaussian mixture. In [4] the skin color cluster is segmented by fitting boundaries directly in the  $C_b C_r$  chromatic space. Color histograms as approximations of probability density functions can be also used, [5, 6].

A face in the Skin Probability Image (SPI) is tracked by determining its position and spatial extent. The mass center of the distribution is used as an estimation of the position [6, 10]. Different techniques to estimate the size have been proposed: threshold on the SPI [1], a function of the zeroth order moment of the distribution [6], a function of its standard deviation [3], boundary detection [4] or combination of the previous ones [7].

A search window is applied to limit the processing range. Each frame it is updated in location and size according to a dynamic model. These models range from the

assumption that there is no movement between frames [6, 3] to the application of Kalman filtering [1, 8]. The color model is also updated to compensate lighting changes [3].

We need a system suitable for real time face tracking to known location, size and pose every frame. The system should be capable to deal with the problems involving people tracking in indoor environments such as: mobile camera, dynamic lighting changes, complex motion dynamics or target overlapping.

In this work we propose a tracking algorithm based on target modeling and projecting to a probability image. The main contribution is a new size and orientation estimation technique from the statistical distribution model complemented with a robust color model update. It has been applied to indoor human face tracking with a hue model which is defined a priori and adaptive. The paper emphasizes the application of the method to human face tracking although it pretends to be general enough to be applicable to any object tracking

The rest of the paper is organized as follows: In the next section the different aspects and structure of the tracking system are described. Section 4 describes the technique to estimate target location and spatial extent. The color model update is shown in section 5. Results of its performance in the presence of the problems described before are presented in section 6 and conclusions are drawn in section 7.

## 2 System Overview

To track a face three characteristics must be determined: face location, spatial extent (size and orientation) and color model. Therefore a target face is modeled by an elliptical region, which comprises its location and spatial extent, and a skin tone model; a one-dimensional hue histogram, used as an approximation of the skin probability density function [5, 6]:

$$p(x_{ij}|skin) \propto hist(H(x_{ij})) . \quad (1)$$

### 2.1 Tracking Initialization

Face hue levels, mainly skin tone, can be modeled a priori and offer the main cue to initially locate target faces on the scene. In this step a generic skin model is used, its values are taken from pixels selected by a statistical segmentation algorithm [5] applied on a face database containing 1500 different persons, with different skin tones and lighting conditions. The algorithm is based on the techniques described in [4]:

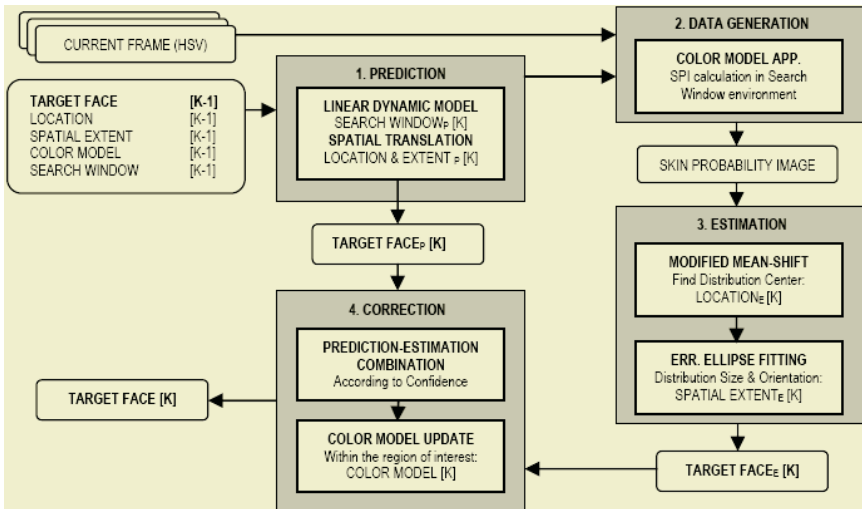
1. Lighting compensation is applied to remove undesired color bias in input image.
2. Skin-like pixels are segmented using the generic tone model and an adaptive threshold.
3. Candidate areas are validated by detecting facial features using chrominance maps for mouth and eyes.

An ellipse is fitted to each of the detected face. The starting area of the face is smaller than the actual one as our elliptical model will adapt itself in size and orientation until taken over the entire facial area in successive iterations, as shown in figure 2.

## 2.2 Tracking Stages

The face tracking system consists of four stages as depicted in figure 1:

1. Prediction: according to the face location in previous frames and a dynamic model, the new position of the search window is predicted. With high frame rates, above 15 per second, the center of the distribution in the previous frame is a valid approximation. However to improve robustness a second order lineal model of constant acceleration is applied. Window size is not modified as its variation follows a dynamic hard to model.
2. Data generation: the image is transformed to the HSV color space [9] and it is used to calculate de SPI image in an area surrounding the search window.
3. Estimation: Target Feature Extraction. The distribution center and spatial extent are computed from the SPI by applying an iterative algorithm to find the distribution center and by fitting an ellipse. It is explained in detail in next section.
4. Correction: predicted and estimated data are linearly combined according to the target characterization process confidence measure. A wide range of face variations is allowed so the estimation confidence is not easy to compute. We have noted that the best combination method is to give preference to extracted data except in clear error situations such as target overlapping or occlusions when there is a bad measure. The hue model for the face target is updated once the final elliptical model is in place; pixels are selected according to their similarity with the previous model.



**Fig. 1.** The different processing steps for a given frame,  $K$ , in the tracking system.  $E$  and  $P$  stand for Estimation and Prediction respectively

## 3 Target Feature Extraction

It is initially based on the CAMShift algorithm idea [6] to compute the center of the probability distribution of a target face. The main innovation is in the estimation of

the spatial extent of the target. It does not require scene dependant thresholds as the one done in CAMShift and also computes the orientation of the target.

### 3.1 Skin Probability Image Generation

The input frame pixels are projected onto the skin tone probability space according to the model of a given target. This is done once for each target, i.e. there is a different SPI for each face. The SPI is zero outside a neighborhood of the search window. In the region of interest the values of the SPI pixels are taken from the hue histogram of the target used as a look up table for the corresponding input frame hue pixels.

As explained in [6] there is a problem when using the HSV space. Hue measure in pixels with corresponding low saturation or high brightness is not reliable. Therefore saturation and brightness planes are also taken in consideration when computing the SPI and these pixels are ignored in the SPI computation.

### 3.2 Target Location

The center of the skin probability distribution is taken as the target location. To locate this position a variation of a mean-shift algorithm [10, 11] is used. Our algorithm is based on the CAMShift algorithm [6] which applies iteratively mean-shift on a size-varying search window until convergence, successfully avoiding local minima.

In CAMShift the search window size is an empirical function of the zeroth moment (weighted area or mass) of the distribution. It also depends on scaling parameters which are scene dependant. This is a problem when lighting modifies the SPI values thus changing the zeroth moment, which results in incorrect window sizing.

We have modified the iterative process of CAMShift so the distribution spatial extent is not estimated until the next step and local minima are still avoided:

1. Center the search window at the predicted face location in the SPI.
2. Set search window size a constant percentage bigger than current one.
3. Compute the mass center of the distribution within the new window.
4. If the location computed in 3 and the window center in 2 do not match, center the search window at the first one and go to step 2.

### 3.3 Target Size and Orientation

The spatial extent of the target face is estimated by fitting an ellipse [12] to the distribution representing the face in the skin probability space.

The distribution orientation is inferred from the main variation modes extracted from its covariance matrix. The covariance matrix can be splitted into independent components made up of the eigenvectors and variances given by eigenvalues. Given a covariance matrix  $H$  with eigenvalues  $v_1$  and  $v_2$ , the orientation:

$$\theta = \tan^{-1} \left( \frac{v_1 - H_{1,1}}{H_{1,2}} \right) - \frac{\pi}{2} . \quad (2)$$



To compute the size we apply the estimation of the distribution uncertainty ellipses (also called error ellipses) for a given confidence interval:

$$\text{MajorAxis} = \sqrt{v_1} \cdot \rho \quad , \quad (3)$$

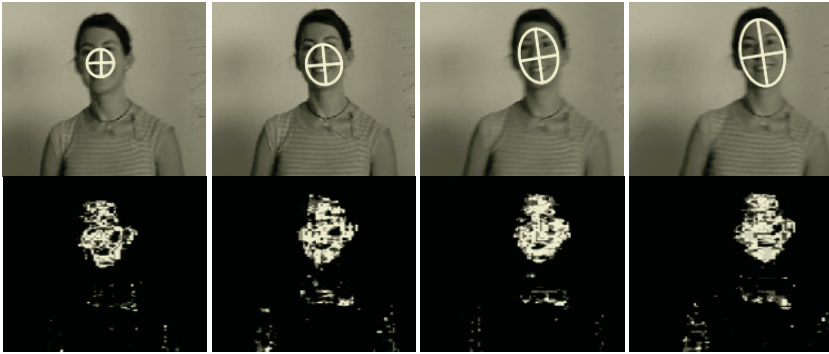
$$\text{MinorAxis} = \sqrt{v_2} \cdot \rho \quad , \quad (4)$$

$$\rho = \sqrt{-2 * \log(1 - P)} \quad , \quad (5)$$

where  $\rho$  is a scaling factor. This defines the axis length of an ellipse such that there is  $P$  probability of being inside it. With a confidence of 99% ( $P=0.99$ ) the selected area contains all relevant information of the distribution. To apply this result we are assuming that the distribution is a bi-dimensional gaussian. It is a good approximation given the correspondence found between the ellipse and the distribution boundaries.

This measuring algorithm is limited to targets whose distribution in the probability space is somewhat elliptical or gaussian. Therefore it is very useful for face modeling as a face boundary can be represented by an ellipse as shown in figure 2.

The fitting process is very robust against close distractors and outliers; with the selected confidence the region of interest is defined as the minimum distribution area containing an ellipse. All nearby values are ignored. The measured size and orientation depend on the covariance matrix, not on the particular values of the distribution. As a result elliptical region is not affected by values with low probability within the target due to shadows or sudden lighting changes; the target area is extrapolated from the distribution variation modes.



**Fig. 2.** First frames of a video sequence and their corresponding SPI's. Estimated face location and axes orientation are depicted on the frames. From its initial placement the face model is able to match face size and orientation in three or four frames and track their variations over time. The elliptical area used to update the color model is also shown

## 4 Robust Color Model Update

The skin tone model, hue histogram, is updated each frame to compensate lighting changes. The model can not be totally remade each frame as it would be too noisy. From previous works we know that the best alternative is a linear combination of the new measured histogram and the previous one:

$$hist[K + 1] = R \cdot hist[k - 1] + (1 - R) \cdot hist_{measured}[K], \quad (6)$$

factor  $R$  is a memory factor, which controls the update speed. It is not critical and any value between 0.2 and 0.8 works.

The main innovation in the model update is the selection of the face pixels to estimate the new histogram; all pixels within a 99% error ellipse of the distribution. The selection is done by Mahalanobis distance, so all pixels within the region of interest are used, regardless of its hue value. A face model starts with only skin tone values but will evolve to represent all hue information in the target as seen in figure 3.



**Fig. 3.** Color model update results: SPI for frames 1, 3 and 5 in a video sequence. Each step the face region contains higher probability levels as the model evolves from the general one

## 5 Experimental Results in Different Tracking Situations

All test video sequences are sized 320x240 pixels and recorded at 15 fps. with a mobile camera and an adaptive iris. The elliptical model is represented on each face target.

Note that target initial locations are defined approximately on target centers and the convergence area is either the face area or the face and neck one, depending on particular lighting and target skin tone. The initial scene conditions are not important as long as there is enough information to adjust the elliptical models to each one. Once the model is in place, the generic skin color model evolves to the target particular one and is capable of following its changes.



**Fig. 4.** The elliptical model tracks target face size and orientation as they change over time even in presence of complex target and camera movement dynamics



**Fig. 5.** Tracking in the case of severe lighting changes. Thanks to the color model update and the robustness against changes in the SPI values of the ellipse fitting algorithm, target face location and spatial extent are preserved. This video sequence shows three main different lighting conditions. The system is also able to track the target during the transitions



**Fig. 6.** Tracking multiple targets. The system is able to track multiple close target faces without confusion between them or with distracters (arms for instance). Simple face overlapping and occlusions are resolved thanks to the linear model prediction and estimation correction

## 6 Conclusions

We have presented a tracking algorithm based on probability distribution tracking. It has been successfully applied to face tracking by using a skin tone model. Our main contributions are an ellipse fitting step to add precision to the estimation of the spatial extent of the target and a robust model update that maintains the tracking under varying lighting conditions.

So far the model is limited to a one-dimensional tone model. The next step is to track using the information contained in all three image planes and pixel neighbor-

hood relationship within the target. The elliptical model fitting can also be upgraded by using multiple ellipses, i.e. modeling the target distribution by a gaussian mixture.

## Acknowledgements

J.J. Gracia Roche is supported by a Phd. Grant B156/2004 from the Diputación General de Aragón. This work is also supported by Grant TIC2003-08382-C05-05 from the Spanish Ministry of Sciences and Technology.

## References

1. Yang, J., Waibel, A., Tracking Human Faces in Real-Time. CMU-CS-95-210. 1995.
2. Störring, M., Andersen, Granum, E.: Skin Colour Detection under Changing Lighting Conditions. In 7<sup>th</sup> Symposium on Intelligent Robotics Systems. 1999.
3. Raja, Y., McKena, S., Gong, S.: Tracking and Segmenting People in Varying Lighting Conditions using Colour. In Proceedings of FG' 98. Nara, Japan. April 14-16, 1998.
4. Hsu, R., Mottaleb, M.A., Jain, A.K.: Face Detection in Color Images. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (5), (2992) 696-706. 1998.
5. C. Orrite, E. Bernués, J. Gracia, I. Gil.: Face detection and recognition in a video sequence. In Proceedings of SPIE. Vol. 5404, Biometric Technology for Human Identification. Orlando (USA). April, 2004.
6. Bradsky, G. R.: Computer Vision Face Tracking for Use in a Perceptual User Interface. In Intel Technology Journal Q2. 1998.
7. Nait-Charif, H., McKenna, S.: Head Tracking and Action Recognition in a Smart Meeting Room. In Proceedings 4th IEEE International Workshop on PETS. Graz, Austria, 2003.
8. Kalman, R.: A New Approach to Linear Filtering and Prediction Problems. In Transactions of the ASME--Journal of Basic Engineering, 82. D. 35-45. 1960
9. Bourgin, D.: Color Space FAQ. In [www.neuro.sfc.keio.ac.jp/~aly/polygon/info/color-space-faq.html](http://www.neuro.sfc.keio.ac.jp/~aly/polygon/info/color-space-faq.html)
10. Comaniciu, D., Armes, V., Meer, P.: Real-Time Tracking of Non-Rigid Objects using Mean Shift. In IEEE CVPR 2000.
11. Cheng, Y.: Mean Shift, Mode Seeking and Clustering. In IEEE Transactions on Pattern Analysis and Machine Intelligence. 17:190-799. 1995.
12. Ribeiro, M.I.: Gaussian Probability Density Functions: Properties and Error Characterization. Institute for Systems and Robotics. Lisboa Portugal. In [www.omni.isr.ist.utl.pt/~mir/pub/probability.pdf](http://www.omni.isr.ist.utl.pt/~mir/pub/probability.pdf)

# Head Gesture Recognition Based on Bayesian Network

Peng Lu, Xiangsheng Huang, Xinshan Zhu, and Yangsheng Wang

Institute of Automation, Chinese Academy of Sciences  
Beijing 100080, China  
{plu,wys}@email.pattek.com.cn

**Abstract.** Head gestures such as nodding and shaking are often used as one of human body languages for communication with each other, and their recognition plays an important role in the development of Human-Computer Interaction (HCI). As head gesture is the continuous motion on the sequential time series, the key problems of recognition are to track multi-view head and understand the head pose transformation. This paper presents a Bayesian network (BN) based framework, into which multi-view model (MVM) and the head gesture statistic inference model are integrated for recognizing. Finally the decision of head gesture is made by comparing the maximum posterior, the output of BN, with some threshold. Additionally, in order to enhance the robustness of our system, we add the color information into BN in a new way. The experimental results illustrate that the proposed algorithm is effective.

## 1 Introduction

With the ever increasing role of computers in society, HCI has become an increasingly important part of our daily lives. Gesture recognition plays an important role in the development of HCI since it provides a natural and efficient interface to computers. Among gestures, head nod and shake are very common and used often, so gesture detection is basic to a visual understanding of human responses.

Up to now, many methods for gesture recognition have been proposed such as syntactical analysis, neural based approach, hidden Markov model(HMM) based recognition, especially in head gestures recognition [1][2][3][4][6]. In [6], a real-time head gesture detector was presented, but an extra hardware was needed. In [4], there is no extra hardware needed and some results are got. But just using color information may result in some problems in bad illumination scenes. The features, which have more statistical significance, should be added to the system. In [3], the pattern between eyes was used for tracking. But it may fail when the pattern doesn't exist in the image, such as rotating head in large range.

In this paper, Bayesian network is introduced for head gesture recognition. Because BN [7] allows one to learn about the causal relationships and predict the future, it is useful when we are trying to understand the head gesture recognition process. As head gesture is the continuous motion on the sequential time series, tracking multi-view head is important and fundamental to gesture recognition. Meanwhile the gesture can be described by a set of head poses, thus it is also

necessary for us to revealing head pose transformation in gesture sequences. Once we construct the multi-view tracking model and gesture inference model, the BN can be applied to integrate them. For head pose detection, much work has been done. Harr wavelet like features were used by Li [10] to recognize multi-view head and he gets good result in multi-view head with floatboost algorithm. But it is not trivial to get the difference between similar poses by Haar features. In [11] a set of appearance based maximum likelihood estimators is used to detect head pose and each estimator describes one pose. By using appearance feature we can get the similar measurement between input pattern and five pose patterns (up, down, left, right and frontal). So it is very useful for our multi-view head tracking model. At the same time skin is arguably the most widely used primitive in human image processing research, with applications ranging from face detection and person tracking to pornography filtering, therefore skin information should be added into the tracking model. Currently the most popular method is learning the skin color distribution model from a large number of training set and using this model to detect head position. But such a model is sensitive to camera lens, which easily leads to shift of center of color model. In this paper we proposed a new way to use color in BN (More details can be found in section 3.2). For sequence actions recognition, HMM is the most effective recognition for sequence gesture recognition. As gesture is the continuous motion on the sequential time series, HMM is the most suitable model as the gesture inference model. The presented method is different from the previous work in the following ways. First, a new way of using color information is proposed and it can eliminate the effect of different camera lens. Second, subspace based methods, which was widely used in hand gestures recognition, are applied in our multi-view model for head gesture recognition. Third, BN is introduced to integrate the multi-view model and gesture inference model. And the proposed method is a nature framework for head gesture recognition.

In the remainder of this paper, section 2 introduces the head gesture inference by BN. Section 3 describes the computing and learning of the BN. Some experiments are shown in section 4.

## 2 Inference in the Bayesian Network

Head gesture is made up of a set of head pose, thus for recognizing head gesture, two steps are needed. First, head pose should be inferred from the MVM, which is made up of color model and appearance model. Second, all the inferred results will be used for head gesture recognition. This process can be compactly represented as a Bayesian network shown in Fig.1. Node  $G$  represents head gesture, nod or shake. Node  $E_i$  is a discrete random variable, which denotes the true head pose in the  $i$ th video frame. Its value can be one of the  $S_1, S_2, \dots, S_5$ , which correspond five head pose: up, down, frontal, left, right.  $C_i, D_i$  are the color and appearance measurements respectively.  $T$  is the number of observations which constitute a sequence. No more than one head gesture should be found in this sequence. In our system we use  $T = 12$ , which was found sufficient to detect slow as well as subtle head nods/shakes [6].

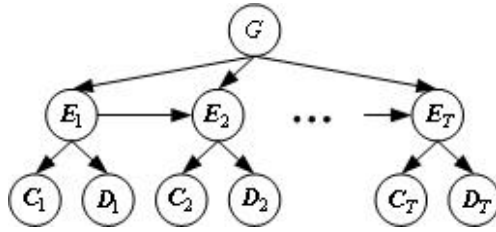


Fig. 1. The Bayesian Network.

From Fig.1 the posterior probability of head gesture given observations of the other variables can be computed as follows:

$$\begin{aligned}
 &P(G|E_1, C_1, D_1, E_2, C_2, D_2 \dots, E_T, C_T, D) \\
 &\propto P(G, E_1, C_1, D_1, E_2, C_2, D_2 \dots, E_T, C_T, D_T) \\
 &= P(E_1, C_1, D_1, E_2, C_2, D_2 \dots, E_T, C_T, D_T|G)P(G) \\
 &= P(C_1, D_1, \dots, C_T, D_T|E_1 \dots, E_T, G)P(E_1, E_2 \dots, E_T|G)P(G)
 \end{aligned} \tag{1}$$

By using conditional independence relationships we can get

$$\begin{aligned}
 &P(C_1, D_1, \dots, C_T, D_T|E_1, E_2 \dots, E_T, G)P(E_1, E_2 \dots, E_T|G)P(G) \\
 &= \prod_{i=1}^T P(C_i|E_i)P(D_i|E_i)P(E_1, E_2 \dots, E_T|G)P(G)
 \end{aligned} \tag{2}$$

Substituting (2) into (1), we have

$$\begin{aligned}
 &P(G|E_1, C_1, D_1, E_2, C_2, D_2 \dots, E_T, C_T, D_T) \\
 &\propto \prod_{i=1}^T P(C_i|E_i)P(D_i|E_i)P(E_1, E_2 \dots, E_T|G)P(G)
 \end{aligned} \tag{3}$$

### 3 Computing and Learning of the Bayesian Network

#### 3.1 Appearance Marginal Likelihood

**Computation of the Appearance Marginal Likelihood.** Based on assumption of a Gaussian distribution, the probability of input pattern  $X$ , which belongs to class  $\Omega$  can be modelled by a multidimensional Gaussian probability density function:

$$P(X|\Omega) = \frac{\exp[-\frac{1}{2}(X - \mu)^T \Sigma^{-1}(X - \mu)]}{(2\pi)^{N/2} |\Sigma|^{1/2}} \tag{4}$$

where  $\bar{X}$  is the mean vector of class  $\Omega$ .  $\Sigma$  is the covariance matrix of class  $\Omega$ .

From equation (4), we have

$$P(D|E) = \frac{\exp[-\frac{1}{2}(C - \mu_i)^T \Sigma_i^{-1}(C - \mu_i)]}{(2\pi)^{N/2} |\Sigma_i|^{1/2}} \tag{5}$$

where  $i = 1, \dots, 5$ , corresponding five head poses.

By given  $\mu_i$  and  $\Sigma_i$ , the appearance likelihood  $P(D|E)$  of each pose  $S$  can be estimated by equation (5). But taking computation into consideration, we use Principal Component Analysis(PCA) to reduce the dimension of  $X$ . In our experiment, the  $P(D|E)$  is approximately estimated by equation (6), more detail about equation (6) can be found in [11].

$$\hat{P}(D|E) = \exp \left[ -\frac{1}{2} \sum_{i=1}^M \frac{y_i^2}{\lambda_i} \right] \exp \left[ -\frac{\epsilon^2(x)}{2\rho} \right] \quad (6)$$

where  $\hat{P}(D|E)$  is the estimation value of  $P(D|E)$ ,  $\epsilon^2(x)$  is the residual error,  $\lambda_i$  is eigenvalue of  $\Sigma$ ,  $M$  is the dimensional of principal subspace,  $N$  is the dimension of total subspace.

**Learning the Appearance Likelihood Model Parameters.** For each head pose the parameters  $\mu_i$  and  $\Sigma_i$  are learned from more than 200 labelled images with different illumination.

### 3.2 Color Marginal Likelihood

In our scheme, we use the region of face in actual frame, which is detected by our Haar-Sobel-like boosting algorithm [13] and aligned by active shape model (ASM) algorithm [9], to create skin color model. At detection stage, Haar and Sobel features are used as feature space. GentleBoost is used to select simple classifiers. Haar features are used to train the first fifteen stages. And then Sobel features are used to train the rest fourteen stages. At alignment stage, ASM is used for face alignment. At color model creating stage, we change the input image from Red, Green, Blue(RGB)color space to HSV space, which separates out hue (color) from saturation (how concentrated the color is) and from brightness. And the color models are created by taking 1D histograms from the H (hue) channel in HSV space. Through this model the input image can be convert into a corresponding head position probability map  $P_c(i, j)$ , which represents the possibility of point( $i, j$ ) belonging to head.

In practice, we assume the skin color distributes as normal distribution  $N(\mu_c, \sigma_c)$ .  $\mu_c$  is the mean value of the H value in face region and  $\sigma_c$  is the variance of H value. Thus we have

$$P_c(i, j) = P(h(i, j)|skin) = \frac{\exp\left[-\frac{(h(i, j) - \mu_c)^2}{2\sigma_c^2}\right]}{\sqrt{2\pi}\sigma_c} \quad (7)$$

where  $h(i, j)$  is the H value at point( $i, j$ ).

We define the color likelihood as follows:

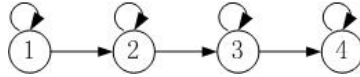
$$P(C|E) = \frac{1}{n_c} \sum_{i, j} P_c(i, j) \quad (8)$$

where  $n_c$  is the scale of the likelihood.



### 3.3 Gesture Marginal Likelihood

**Computation of the Gesture Marginal Likelihood.** Given the observation sequence  $E_1, E_2 \dots, E_T$ , the likelihood  $P(E_1, E_2 \dots, E_T|G)$  can be derived from discrete HMM model, which is represented by  $\lambda = (A, B, \Pi)$ , where  $A = (a_{ij})_{N \times N}$  is the state transition probability distribution matrix,  $B = (b_{jk})_{N \times T}$  is the observation symbol probability distribution matrix and  $E_k$  represents discrete observation symbol,  $\Pi = (\pi_1, \dots, \pi_N)$  is the initial state distribution.



**Fig. 2.** A left to right state transition diagram for a 4-state HMM. For head nod 1,2,3,4 denote frontal, up, frontal, down respectively. For head shake 1,2,3,4 denote frontal, left, frontal, right respectively.

As show in Fig. 2, the observation sequence  $E_1, E_2 \dots, E_T$  and head gesture  $G$  are connected through HMM model  $\lambda = (A, B, \Pi)$ .

$$\begin{aligned}
 P(E_1, E_2, \dots, E_T|G) &= \sum_{all Q} P(E_1, E_2, \dots, E_T|Q, G)P(Q|G) \\
 &= \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(E_1) a_{q_1 q_2} b_{q_2}(E_2) \dots a_{q_{T-1} q_T} b_{q_T}(E_T)
 \end{aligned}
 \tag{9}$$

In practice, the  $P(E_1, E_2, \dots, E_T|G)$  can be computed by Forward-Backward algorithm [12] as follows:

we define  $\alpha_t(i)$  as follows:

$$\alpha_t(i) = P(E_1, E_2, \dots, E_T, q_t = S_i|G)
 \tag{10}$$

1) Initialization:

$$\alpha_1(i) = \pi_i b_i(E_1) 1 \leq i \leq N.
 \tag{11}$$

2) Induction:

$$\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) a_{ij}] b_j(E_{t+1})
 \tag{12}$$

3) Termination:

$$P(E_1, \dots, E_T|G) = \sum_{i=1}^N \alpha_T(j)
 \tag{13}$$

**Learning the Gesture Likelihood Model Parameters.** In our experiment fifty labelled gesture sequences were used for training. The parameters  $(A, B, \Pi)$  can be learned by Baum-Welch algorithm [12] as follows:

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j)
 \tag{14}$$

$$\bar{\pi}_i = \gamma_1(i) \tag{15}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{16}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \tag{17}$$

where  $\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | E_1, E_2, \dots, E_T, G)$ .

### 4 Experimental Result

Two BN are necessary to recognize head gestures and each one describes a gesture. For each BN, there are fifteen parameters,  $\mu_i$   $1 \leq i \leq 5$ ,  $\Sigma_i$   $1 \leq i \leq 5$ ,  $u_c, \sigma_c, A, B, \Pi$ . The  $\mu_i, \Sigma_i$ , are obtained from 200 labelled images, of which the size is  $20 \times 20$ . The  $u_c, \sigma_c$  are obtained from the aligned face region. And the  $A, B, \Pi$  are obtained from fifty labelled gesture sequences. We implemented the algorithm on a P4 machine with 1.3G CPU and the database collection of our experiment was similar to that in [6]. A program agent will ask 5 questions, and the subjects are asked to answer with head nod or head shake.

As an example, Fig. 3 shows the recognition process of head shake. Our algorithm is initialized by the output of Haar-like feature based boosted cascade face detector. Once a frontal face is detected, the color model will be created by the aligned region and the size of head in image can be got. The next twelve frames will be taken out. With Kalman filter the coarse head region in each frame is cropped as an input of BN. MVM will be applied to the input to detect head position and infer the head pose. But instead of make a decision of head pose, we send to HMM the three maximal pose inference results out of five  $S$ . Through BN integrating, we can obtain the maximal a posterior (MAP). Finally, a gesture decision will be made by comparing the MAP with some threshold.

**Table 1.** Recognition results.

In the Lab			
Nod	Shake	Nod Miss	Shake Miss
67	73	7	4
Average Recognition ratio: 92.1%			

Table 1 illustrates the recognition results of the samples in the database and the average recognition ratio of head gesture is 92.1%. From Table 1 we can see

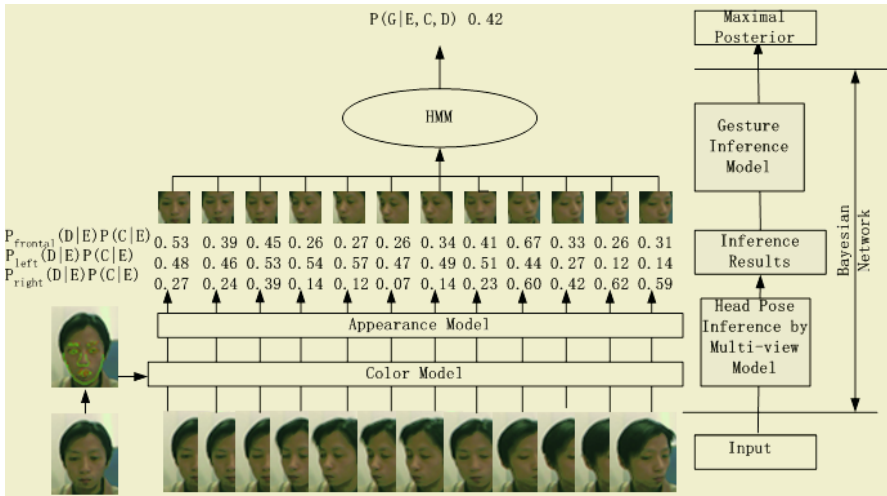


Fig. 3. A recognition process of head shake.

that some head gestures go undetected, but there are no false positives. In the database the persons in some samples have expression, which doesn't affect the recognition of head gesture. The undetected head gestures due to the slight head movement. The outputs of multi-view head tracking model are inaccurate as the appearance features, the base of the tracking model, are not sensitive to these slight movement.

## 5 Summarize

In this paper a unified framework for recognizing head gesture in HCI environment is proposed. The MVM is used for tracking head pose and the HMM is used for inferring the head gesture. Finally, MVM and HMM are integrated into the BN framework. Experimental results show that the proposed framework is feasible and effective. But it doesn't work well in slight head movement. Local feature tracking seems to be a good way to solve the slighter movement. So how to fuse more local features into the proposed framework will be our future works. Comparing with the previous work, the main contribution of this paper is introducing the BN into head gesture recognition, providing a way for multi information (such as appearance and color) fusing. Thus the recognition process of head gesture is more efficient.

## References

1. U.M. Erdem, S. Sclaroff, "Automatic Detection of Relevant Head Gestures in American Sign Language Communication," Proceedings. 16th International Conference on Pattern Recognition, 2002., Volume: 1 , 11-15, Pages:460 - 463 vol.1, Aug. 2002.

2. O. Deniz, A. Falcon, J. Mendez, M. Castrillon, "Useful Computer Vision Techniques for Human-Robot Interaction," ICIAR 2004-International Conference on Image Analysis and Recognition, Porto, Portugal 2004.
3. S. Kawato, and J. Ohya, "Real-time Detection of Nodding and Head-shaking by Directly Detecting and Tracking the 'Between-Eyes'," in Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, 2000.
4. Pei Chi Ng and L.C. De Silva, "Head Gestures Recognition," Proceedings. 2001 International Conference on Image Processing, Volume: 3 , 7-10, Pages:266 - 269 vol.3, Oct. 2001.
5. G. Rigoll, A. Kosmala and M. Schuster, "A new approach to video sequence recognition based on statistical methods," In IEEE Int. Conference on Image Processing (ICIP), pages 839-842, Lausanne, September 1996.
6. A. Kapoor and RW Picard, "A real-time head nod and shake detector," Workshop on Perspective User Interfaces, November 2001.
7. D. Heckerman, "A Tutorial on Learning With Bayesian Networks," Microsoft Research Technical Report, MSR-TR-95-06
8. M. Yang, D. J. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," IEEE Trans. on PAMI, vol. 24, no.1, pp.34-58, Jan. 2002.
9. T.F. Cootes, C.J. Taylor "Statistical Models of Appearance for computer vision," Imaging Science and Biomedical Engineering, University of Manchester, Manchester M13 9PT, U.K. March 8, 2004.
10. S. Li, L. Zhu, Z.Q. Zhang, A. Blake, H.J. Zhang, H. Shum, "Statistical Learning of Multi-View Face Detection," Proc. 7th ECCV, Copenhagen, Denmark, May 2002.
11. B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):696-710, July 1997.
12. L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. IEEE, vol. 77, no. 2, pp. 257-286, 1989
13. P. Lu, X.S. Huang, Y.S. Wang, "A New Framework for Handfree Navigation in 3D Game," Proceedings of the International Conference on CGIV04.

# Detection and Tracking of Face by a Walking Robot\*

Do Joon Jung<sup>1</sup>, Chang Woo Lee<sup>2</sup>, and Hang Joon Kim<sup>1</sup>

<sup>1</sup> Department of Computer Engineering, Kyungpook National Univ., Korea  
{djjung,hjkim}@ailab.knu.ac.kr

<sup>2</sup> Department of Computer Information, Kunsan National Univ., Korea  
leecw@kunsan.ac.kr

**Abstract.** We propose a system for detection and tracking of face in dynamic and changing environments from a camera mounted on a walking robot. The proposed system is based on the principal component analysis (PCA) technique. For the detection of a face, first, we use a skin color information and motion information. Thereafter, we verify that the detected regions are indeed the face using the PCA technique. The tracking of a face is based on the Euclidian distance in eigenspace between the previously tracked face and the newly detected faces. Walking robot control for the face tracking is done in such a way that the detected face region is kept on the central region of the camera screen by controlling the robot motion. The proposed system is extensible to other walking robot systems and gesture recognition systems for human-robot interaction.

**Keywords:** Face Detection, Face Tracking, PCA, Walking Robot

## 1 Introduction

The mobile machines with wheels and crawlers assume simple works and their movable environment is limited. Because of the movable environment limitation, humanoid 2-leg walking robot has been produced such as ASIMO and SDR-4X and so on. One of the goals of building intelligent and interactive machines is to make them aware of the user's presence. Detection and tracking of face from a walking robot is a much more challenging problem as the scene is much more dynamic because of both motion of the camera and that of the user.

In general, there are two kinds of grouping of tracking methods according to their views. Some people group tracking methods as recognition-based tracking and motion-based tracking and the others group them as edge-based tracking and region-based tracking [1].

Recognition-based tracking is really based on the object recognition technique and the performance of the tracking system is limited by the efficiency of

---

\* This work is financially supported by the Ministry of Education and Human Resources Development(MOE) and the Ministry of Commerce, Industry and Energy(MOCIE) through the fostering project of the Industrial-Academic Cooperation Centered University.

the recognition method. Motion-based tracking relies on the motion detection technique, which can be divided into the optical flow method and the motion-energy method.

Edge-based methods track the edges in an image sequence, which are usually boundaries of objects of interest. However, these methods suffer from the changes in color or illumination since boundaries of objects to be tracked have to show a strong edge variation in color or illumination. Moreover, it is difficult to provide reliable results in a case where the background of an image has strong edges. Most of the current work related to this type of method stems from the efforts of Kass et al. on snakes [2]. Many of the recent researches on face tracking are in trouble with the presence of background noise and apt to track an unverified face, for example, arms and hands.

In this paper, we propose a system for detection and tracking of face in dynamic and changing environments from a camera mounted on a walking robot using PCA technique. The proposed system consists of two main steps as depicted in Figure 1: face detection and face tracking. Using two consecutive frames, first, the candidate face regions are verified to determine which region is indeed the face using PCA. Thereafter, the verified face is tracked using the eigen-technique.

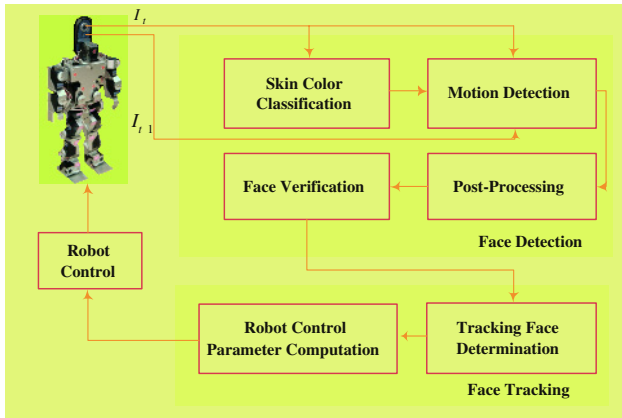


Fig. 1. Architecture of the proposed system.

## 2 Face Detection

In this section, the techniques used to detect faces in the proposed system are introduced. For improving the accuracy of the face detection, we combine several published techniques such as a skin color model [3] and PCA [4, 5].

### 2.1 Skin Color Classification

Detecting pixels with the skin color provides a reliable method for detecting and tracking faces. Since an *RGB* representation obtained by most video cameras not

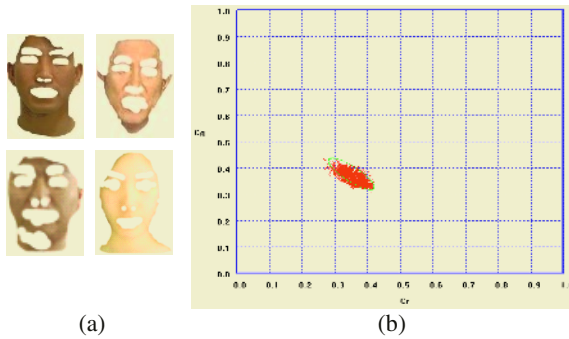
only includes color but also brightness, this color space is not necessarily the best color representation for detecting pixels with skin color. The brightness may be removed by dividing the three components of a color pixel by the intensity. The color distribution of human faces is clustered in a small area of the chromatic color space [6].

When the chromatic  $r$  and chromatic  $g$  of skin pixels from face patch are plotted in  $CrCg$ -space, skin color occupy with elliptical shape. Figure 2(a) are the sample face patches and (b) shows the skin locus of a MPC-C30 CCD camera used in experiment. A simple membership function to the skin locus is a pair of quadratic functions defining the upper and lower bound of the cluster [7]. But, in our experiment, the shape of skin cluster is similar to an ellipse. Therefore, we decide the membership function to the skin locus is an elliptical function as follows:

$$S = \begin{cases} 1, & \frac{(x' - c_x)}{a^2} + \frac{(y' - c_y)}{b^2} < 1, \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad (2)$$

where  $C_x = 0.426$ ,  $C_y = 0.324$ ,  $a = 0.074$ ,  $b = 0.022$ ,  $\theta = 0.226$  (in radian) are computed from the skin cluster in the  $CrCg$ -space.



**Fig. 2.** (a) The sample face patches, (b) Skin locus of MPC-C30 CCD camera in  $CrCg$ -space.

## 2.2 Motion Detection

Although skin color is the most widely used feature, skin color alone is not suitable for the face detection in the case when skin colors appear in the background areas as well as in the human skin areas. This drawback can be effectively removed by using motion information. To be precise, after the skin classification, only those skin color regions are considered, which contain motion. As a result, the combined skin color model with motion information results in a binary image that indicates the foreground (face regions) and background (non-face region). The binary image is defined as

$$M_t(x, y) = \begin{cases} 1, & I_t(x, y) \in S_t \ \& \ |I_t(x, y) - I_{t-1}(x, y)| > \theta_t \\ 0, & \textit{otherwise} \end{cases}, \quad (3)$$

where  $I_t(x, y)$  and  $I_{t-1}(x, y)$  are the intensities of the current frame and previous frame at pixel  $(x, y)$ , respectively.  $S_t$  is a set of the skin color pixels of the current frame and  $\theta_t$  is a threshold value calculated using an adaptive thresholding technique [8]. As a post-processing we simplify the  $M_t$  image using morphological operations and connected component analysis.

### 2.3 Face Verification Using PCA

In a sequence, tracking of the face of interest is difficult because there are many moving objects. Moreover, a process is needed to verify that the moving object is a face or not. For the face verification problem, we use the weight vectors of candidate regions in eigenspace. For the dimensionality reduction of the feature space, we project an N-dimensional candidate face image to the lower-dimensional feature space, called eigenspace or facespace [4, 5]. In eigenspace, each feature component accounts for a different amount of the variation among the face images. To be brief on the eigenspace, let a set of images be  $I_1, I_2, I_3, \dots, I_M$ , which is the N-dimensional column vector of each image and used for constructing the facespace. The average of the training set is defined by  $A = 1/M \sum_{i=1}^M I_i$ . A new set of vectors with zero mean at each dimension is computed as  $\Phi_i = I_i - A$ . To produce the  $M$  orthogonal vectors that optimally describe the distribution of face images, the covariance matrix is originally computed as

$$C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T = YY^T, \quad (4)$$

for  $Y = [\Phi_1 \Phi_2 \dots \Phi_M]$ . Since the matrix  $C$ , however, is  $N \times N$  dimension, determining the N-dimensional eigenvectors and N eigenvalues is an intractable task. Therefore, for the computational feasibility, instead of finding the eigenvectors for  $C$ , we calculate  $M$  eigenvectors  $v_k$  and eigenvalues  $\lambda_k$  of  $[Y^T Y]$ , so that  $u_k$ , a basis set is computed as

$$u_k = \frac{Y \times v_k}{\sqrt{\lambda_k}}, \quad (5)$$

for  $k = 1, \dots, M$ . Of the  $M$  eigenvectors, the  $M'$  significant eigenvectors are chosen as those with the largest corresponding eigenvalues. For  $M$  training face images, the feature vectors  $W_i = [w_1, w_2, \dots, w_{M'}]$  are calculated as

$$w_k = u_k^T \Phi_i, \quad k = 1, \dots, M'. \quad (6)$$

To verify the candidate face region is indeed the face image, the candidate face regions are also projected into the trained eigenspace using equation (6). The projected regions are verified using the minimum distances of the detected regions with the face cluster and the non-face cluster according to equation (7).

$$\min(\|W_k^{\text{candidate}} - W_{\text{face}}\|, \|W_k^{\text{candidate}} - W_{\text{non-face}}\|), \quad (7)$$



where  $W_k^{candidate}$  is the  $k$ th candidate face region in trained eigenspace, and  $W_{face}, W_{non-face}$  are the center coordinate of the face cluster and non-face cluster in trained eigenspace respectively, and the Euclidean distance measure is used.

### 3 Face Tracking

Among the newly detected faces, the face to be tracked in the next image sequence is determined by using a distance measure in the eigenspace. For tracking of the face, the Euclidian distance between the feature vectors of the tracked face and those of the  $K$  newly detected faces is calculated as

$$obj = \arg \min_k \|W_{old} - W_k\|, \quad k = 1, \dots, K. \quad (8)$$

After the determination of the face region, central position of the face region is verified that the central position is inside of central region of screen or not. If the face region is located out of central region, face size and face direction are calculated. Thereafter, the distance between the center of the detected face region and the center of the screen is calculated as

$$dist_t(face, screen) = Face_t(x, y) - Screen\left(\frac{height}{2}, \frac{width}{2}\right), \quad (9)$$

where  $Face_t(x, y)$  is the center of the detected face region at  $t$  time and  $Screen(height/2, width/2)$  is the center of the screen. Using the face size, face direction and distance vector, horizontal and vertical motion is controlled. The camera control is done in such a way that the detected face region is kept on the central region of the camera screen by controlling the robot motion. We controlled the robot motion using predefined robot actions such as “walk forward”, “walk backward”, “move right”, “move left”, “turn right”, “turn left”.

### 4 Experimental Results

The experimental environment was the laboratory room where possible noises were existed and the lighting condition was changing. Figure 3 shows the setup and the interface of the proposed system.

For tracking of face, the system control a mobile robot. The controlled results are appeared as a action of robot. The used robot, *KHR-1* can move to various orientations using a servo control board, *RCB-1* equipped in a robot. The *RCB-1* is operated by HeartToHeart1.0 software which is shown Figure 4(b). To operate a robot, we control the HeartToHeart1.0. First, we make robot actions by motion generator in HeartToHeart1.0. Thereafter, we assign a number to each action and memorize the assigned action by motion controller. Given a action number, a robot controller controls the robot action.

In the experiments, a user is standing before a mobile robot with complex background. The images are obtained from the camera mounted on the *KHR-1*.

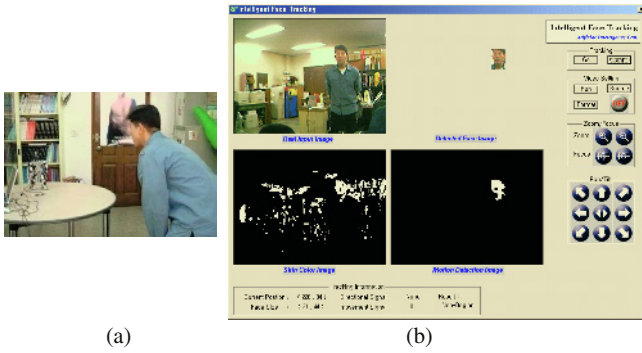


Fig. 3. The setup of the proposed system: (a) the setup, (b) the interface.

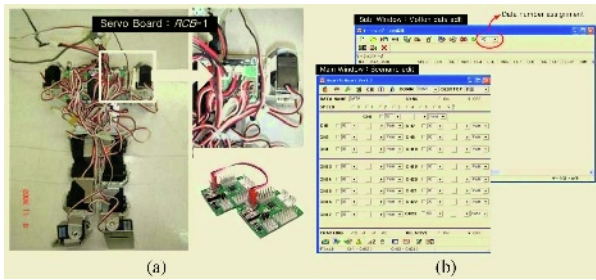


Fig. 4. A robot controller: (a) control board:RCB-1, (b) Graphic user interface (GUI): HeartToHeart1.0.



Fig. 5. The part of the training images for eigenspace construction.

The used robot is *KHR-1* which can controlled servo motor. For performance evaluation, the proposed system was tested on 20 different test sequences and the training set consists of 13 individuals at 5 different head orientations. Figure 5 shows the part of the training images which is used in construction of the eigenspace.

The analysis of a set of images captured during the experiment revealed that the correct rate of face verification was 82.3% in an average.

$$\begin{aligned}
 & \text{Face verification rate} \\
 &= \frac{\text{Number of correctly verified faces}}{\text{Number of images verified as true face}}, \tag{10}
 \end{aligned}$$

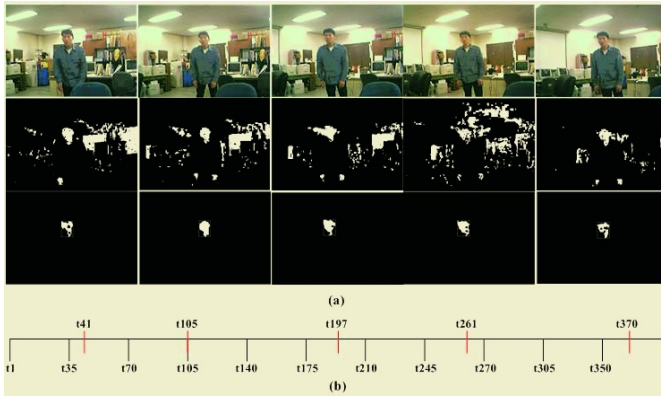
$$\begin{aligned} & \text{Non-face verification rate} \\ & = \frac{\text{Number of correctly verified images as non-faces}}{\text{Number of images verified as non-face}}. \end{aligned} \quad (11)$$

In Table 1, the correct verification rates that the face region is verified as the face and the non-face region is verified as the non-face are shown.

**Table 1.** Face verification rates.

Users	Face	Non-Face	Total
User 1	82.1%	77.6%	79.85%
User 2	83.5%	79.3%	81.4%
User 3	85.2%	81.1%	83.15%
User 4	84.7%	80.5%	82.6%
User 5	86.5%	82.5%	84.5%
Total	84.4%	80.2%	82.3%

In Figure 6, we show the results of the proposed system in which face tracking results are in (a), face detected instants are in (b). In the Figure 6 (a), top row shows the input images, middle row shows skin regions of input images and bottom row shows the tracked face region.



**Fig. 6.** The results of the proposed system: (a) face tracking results, (b) face detected instants.

## 5 Conclusions

In this paper, we proposed a system for detection and tracking of face in dynamic and changing environments from a camera mounted on a walking robot. The proposed system was operated in real-time and performed in two main steps: face detection and face tracking. In the input video sequences, first, we detected the face regions using multi-cues such as color, motion information and PCA. The tracking of a face is done in such a way that the detected face region is kept on the central region of the screen through controlling a walking robot motion.

A robustness of the proposed system in possibly noisy environment was shown in the experimental results. The proposed system is extensible to other walking robot systems and gesture recognition systems for human-robot interaction. For the future work, we will have more experiments of the proposed system in other big size walking robot environments.

## References

1. Shearer, K., Wong, K.D., Venkatesh, S.: Combining multiple tracking algorithms for improved general performance. *Pattern Recognition*, Vol. 34 (2001) 1257–1269
2. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. *Int. J. Computer Vision*, Vol. 1 (1996) 321–331
3. Hsu, R.L., Abdel-Mottaleb, M., Jain, A.K.: Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24 (2002) 696–706
4. Turk, M., Pentland, A.: Face Recognition Using Eigenfaces. *IEEE Proceedings CVPR '91*, (1991) 586–591
5. Guan, A.X., Szu, H.H.: A local face statistics recognition methodology beyond ICA and/or PCA. *International Joint Conference on Neural Network*, (1999) 1016–1021
6. Yang, J., Waibel, A.: A Real-Time Face Tracker. *IEEE Workshop on Application of Computer Vision*, (1996) 142–147
7. Soriano, M., Martinkauppi, B., Huovinen, S., Laaksonen, M.: Adaptive skin color modeling using the skin locus for selecting training pixels. *Pattern Recognition*, Vol. 36 (2003) 681–690
8. Habili, N., Moini, A., Burgess, N.: Automatic Thresholding for Change Detection in Digital Video. *SPIE Proceedings on Visual Communications and image Processing*, Vol. 4067 (2000) 133–142

## Part VI

# Human Activity Analysis

# Appearance-Based Recognition of Words in American Sign Language

Morteza Zahedi, Daniel Keysers, and Hermann Ney

Lehrstuhl für Informatik VI – Computer Science Department  
RWTH Aachen University – D-52056 Aachen, Germany  
{zahedi,keysers,ney}@informatik.rwth-aachen.de

**Abstract.** In this paper, we present how appearance-based features can be used for the recognition of words in American sign language (ASL) from a video stream. The features are extracted without any segmentation or tracking of the hands or head of the signer, which avoids possible errors in the segmentation step. Experiments are performed on a database that consists of 10 words in ASL with 110 utterances in total. These data are extracted from a publicly available collection of videos and can therefore be used by other research groups. The video streams of two stationary cameras are used for classification, but we observe that one camera alone already leads to sufficient accuracy. Hidden Markov Models and the leaving one out method are employed for training and classification. Using the simple appearance-based features, we achieve an error rate of 7%. About half of the remaining errors are due to words that are visually different from all other utterances.

## 1 Introduction

Deaf people need to communicate with hearing people in everyday life. To facilitate this communication, systems that translate sign language into spoken language could be helpful. The recognition of the signs is the first step in these systems. Several studies on gesture and sign language recognition have been published. These publications can be separated into three categories according to the signs they try to recognize.

1. In the first category, researchers propose methods to recognize static hand postures or the sign language alphabet [1–4]. They use images of the hands and extract feature vectors according to the static information of the hand shape. This approach cannot recognize the letters of the sign language alphabet that contain local movement made by the wrist, knuckles, or finger joints, as e.g. the sign for ‘j’ in American sign language (ASL).
2. The researchers in the second category [5, 6] collect sequential feature vectors of the gestures and, using the dynamic information, recognize letters with local movement, too. In these approaches, only movement due to changing hand postures is regarded, while path movement is ignored (movement made primarily with the shoulder or elbow).

3. The third category of researchers try to recognize sign language words and sentences [7–10]. In addition to local movement of the hands, signing includes also path movement of the hands. Therefore, most systems employ segmentation and tracking of the hands.

Most researchers use special data acquisition tools like data gloves, colored gloves, location sensors, or wearable cameras to extract features. Some researchers of the first and second category use simple stationary cameras [1, 2] without any special data acquisition tools but their images only show the hand. Skin color segmentation allows them to perform a perfect segmentation. In the third category because of the occlusion between hands and the head of the signer, segmentation based on skin color is very difficult. Instead of gloves, some researchers use different methods. For example in [9] the camera is placed above the signer in front of him. Then in the images captured by this camera the occlusion between the hands and head of the signer is decreased. These methods or special tools may be difficult to use in practical situations.

In contrast to existing approaches, our system is designed to recognize sign language words using simple appearance-based features extracted directly from the frames captured by standard cameras. This means that we do not rely on complex preprocessing of the video signal. Using only these simple features, we can already achieve a satisfactory accuracy. Those utterances of the data that are still misclassified are due to a strong visual difference from the other utterances in the database. Since our data are based on a publicly available collection of videos, other research groups are able to compare their results to those presented in this paper. Furthermore, our system is designed to work without any segmentation or tracking of the hands. Because we do not rely on an intermediate segmentation step, the recognition can be expected to be more robust in cases where tracking and segmentation are difficult.

## 2 Database

The National Center for sign language and Gesture Resources of the Boston University published a database of ASL sentences [11]. Although this database has not been produced primarily for image processing research, it consists of 201 annotated video streams of ASL sentences.

The signing is captured simultaneously by four standard stationary cameras where three of them are black/white and one is a color camera. Two black/white cameras, placed towards the signer’s face, form a stereo pair and another camera is installed on the side of the signer. The color camera is placed between the stereo camera pair and is zoomed to capture only the face of the signer. The movies published on the internet are at 30 frames per second and the size of the frames is  $312 \times 242$  pixels<sup>1</sup>. We use the published video streams at the same frame rate but we use only the upper center part of size  $195 \times 165$  pixels because parts of the bottom of the frames show some information about the frame and the left and right border of the frames are unused.

<sup>1</sup> <http://www.bu.edu/asllrp/ncslgr.html>

**Table 1.** List of the words and number of utterances in the BOSTON10 database.

Word	Number of utterances
CAN	17
BUY	15
CAR	15
BOOK	13
HOUSE	11
WHAT	10
POSS (Possession)	9
WOMAN	8
IX “far” (Pointing far)	7
BREAK-DOWN	5
Sum	110

**Fig. 1.** The signers as viewed from the two camera perspectives.

To create our database for ASL word recognition that we call BOSTON10, we extracted 110 utterances of 10 words from this database as listed in Table 1. These utterances are segmented manually.

In BOSTON10, there are three signers: one male and two female signers. All of the signers are dressed differently and the brightness of their clothes is different. We use the frames captured by two of the four cameras, one camera of the stereo camera pair in front of the signer and the other lateral. Using both of the stereo cameras and the color camera may be useful in stereo and facial expression recognition, respectively. Both of the used cameras are in fixed positions and capture the videos in a controlled environment simultaneously. In Figure 1 the signers and the views of the cameras are shown.

### 3 Appearance-Based Features

In this section, we briefly introduce the appearance-based features used in our ASL word recognition. The definition of the features is based on basic methods of image processing. These features are directly extracted from the images. We denote by  $X_t(m, n)$  the pixel intensity at position  $(m, n)$  in the frame  $t$ .



**Original images (OI).** We can transfer the matrix of an image to a vector  $x_t$  and use it as a feature vector. To decrease the size of the feature vector, we use the original image down-sampled to  $13 \times 11$  pixels denoted by  $X'_t$ .

$$x_t(i) = X'_t(m, n), \quad i = 13 \cdot n + m$$

**Skin intensity thresholding (SIT).** To ignore background pixels, we use skin intensity thresholding. This thresholding is not a perfect segmentation and we cannot rely on it easily for tracking the hands because the output of this thresholding consists of the two hands, face and some parts of the signer's clothes.

$$\tilde{x}_t(i) = \begin{cases} x_t(i) & : x_t(i) > \Theta \\ 0 & : \text{otherwise} \end{cases}$$

Where  $\tilde{x}_t$  is the feature vector at time  $t$  with the brightness threshold  $\Theta$ .

**First derivative (FD).** This feature measures the rate of change between the successor frame and the predecessor frame and is denoted by  $\hat{x}_t$ .

$$\hat{x}_t(i) = \tilde{x}_{t+1}(i) - \tilde{x}_{t-1}(i)$$

**Positive first derivative (PFD).** This feature vector consists of positive members of the FD feature vector. The feature vector has the information of some pixels of the image that in the predecessor frame do not belong to the skin intensity values but in the successor frame they are in the skin intensity values.

$$\hat{x}_t(i) = \begin{cases} \tilde{x}_{t+1}(i) - \tilde{x}_{t-1}(i) & : \tilde{x}_{t+1}(i) - \tilde{x}_{t-1}(i) > 0 \\ 0 & : \text{otherwise} \end{cases}$$

**Negative first derivative (NFD).** In contrast to the PFD feature vector, the NFD feature vector at time  $t$  indicates the intensity of the pixel is decreasing. This feature has information of some pixels of the image that in the predecessor frame are in the skin intensity values but in the successor frame hands or face of the signer leave that region and they do not belong to the skin intensity values.

$$\hat{x}_t(i) = \begin{cases} \tilde{x}_{t+1}(i) - \tilde{x}_{t-1}(i) & : \tilde{x}_{t+1}(i) - \tilde{x}_{t-1}(i) < 0 \\ 0 & : \text{otherwise} \end{cases}$$

**Absolute first derivative (AFD).** This feature consists of the combined information of the PFD and NFD feature vectors by using the absolute value of the temporal difference images.

$$\hat{x}_t(i) = |\tilde{x}_{t+1}(i) - \tilde{x}_{t-1}(i)|$$

**Second derivative (SD).** The information related to the acceleration of the changes or movements can be found in the SD feature vector.

$$\hat{x}_t(i) = \tilde{x}_{t+1}(i) - 2 \cdot \tilde{x}_t(i) + \tilde{x}_{t-1}(i)$$

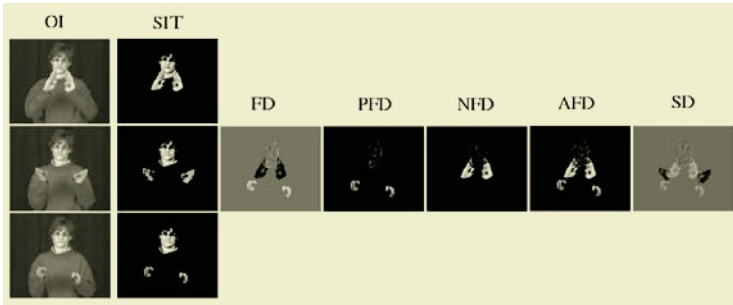


Fig. 2. Examples for the appearance-based features.

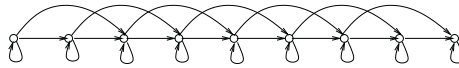


Fig. 3. The topology of the employed HMM.

We apply the skin intensity thresholding to the original frames and then extract derivative feature vectors. Some examples of features after processing are shown in Figure 2.

The feature vectors defined above can be concatenated to provide new feature vectors with more information. In addition, to increase the information extracted from the signer, we may use the frames of two cameras. One of the cameras is installed in front of the signer and the second one is fixed at one side. We concatenate the information of the frames captured simultaneously by these cameras. We weight the features extracted by the two cameras because we have more occlusion of the hands in images captured by the lateral camera.

## 4 Decision Making

The decision making of our system employs Hidden Markov Models (HMM) to recognize the sign language words<sup>2</sup>. This approach is inspired by the success of the application of HMMs in speech [12] and also most sign language recognition systems [7–10]. The recognition of sign language words is similar to spoken word recognition in the modelling of sequential samples.

The topology of the HMM is shown in Figure 3. There is a transition loop at each state and the maximum allowable transition is set to two. We consider one HMM for each word  $w = 1, \dots, W$ . The basic decision rule used for the classification of  $\hat{x}_1^T = \hat{x}_1, \dots, \hat{x}_t, \dots, \hat{x}_T$  is:

$$\begin{aligned} r(\hat{x}_1^T) &= \arg \max_w (Pr(w|\hat{x}_t)) \\ &= \arg \max_w (Pr(w) \cdot Pr(\hat{x}_t|w)) \end{aligned}$$

<sup>2</sup> Some of the code used in feature extraction and decision making is adapted from the LTI library which is available under the terms of the GNU Lesser General Public License at <http://ltilib.sourceforge.net>.

where  $Pr(w)$  is the prior probability of class  $w$  and  $Pr(\hat{x}_t|w)$  is the class conditional probability of  $\hat{x}$  given class  $w$ . the  $Pr(\hat{x}_t|w)$  is defined by:

$$Pr(\hat{x}_t|w) = \max_{s_1^T} \prod_{t=1}^T Pr(s_t|s_{t-1}, w) \cdot Pr(\hat{x}_t|s_t, w)$$

where  $s_1^T$  is the sequence of states and  $Pr(s_t|s_{t-1}, w)$  and  $Pr(\hat{x}_t|s_t, w)$  are the transition probability and emission probability, respectively. The transition probability is calculated by simple counting. We use the Gaussian and Laplace function as emission probability distributions  $Pr(\hat{x}_t|s_t, w)$  in the states. To estimate  $Pr(\hat{x}_t|s_t, w)$  we use the maximum likelihood estimation method for the Gaussian and Laplace functions, i.e. standard deviation and mean deviation estimation, respectively. The number of states for the HMM of each word can be determined in two ways: minimum and average sequence length of the training samples. Mixture densities with a maximum number of five densities are used in each state.

We use the Viterbi algorithm to find the sequence of the HMM. In addition to the density-dependent estimation of the variances, we use pooling during the training of the HMM which means that we do not estimate variances for each density of the HMM, but instead we estimate one set of variances for all densities in each state of the model (state-dependent pooling) or for all densities in the complete model (word-dependent pooling).

The number of utterances for each word is not large enough to separate them into training and test sets, therefore we employ the leaving one out method for training and classification. That is, we separate each utterance as a test sample, train the HMM of each word with the remaining utterances, and finally classify the test utterance. We repeat this process for all utterances in the database. The percentage of the misclassified utterances is the error rate of the system.

## 5 Experimental Results

First, we choose the down-sampled original image after skin intensity thresholding and employ the HMM classifier to classify words of the database. The results of this classification using the Gaussian distribution with different sequence lengths and pooling are shown in Table 2. Using word-dependent pooling gives better results than state-dependent pooling or density-dependent estimation of the variances. Using the Laplace distribution, the performance of the classifier is similar to these results but the Gaussian distribution performs better.

We employ an HMM of each word with the length of the minimum and average sequence length of the training samples. As it is shown in Table 2, neglecting other parameters, the shorter HMMs give better results. This may be due to the fact that the database is small and if the HMM has fewer states, the parameters of the distribution functions will be estimated better. In informal experiments with shorter HMMs the accuracy of the classifier could not be improved.

We use other appearance-based features in the HMM with the Gaussian emission probability distribution. The length of the HMM for each word is minimum

**Table 2.** Error rate (%) of the classifier with different pooling and length parameters.

Sequence length	Pooling		
	Word-dependent	State-dependent	Density-dependent
Minimum seq. length	<b>7</b>	8	<b>7</b>
Average seq. length	14	15	17

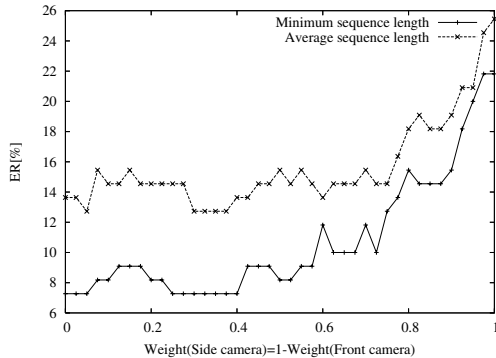
**Table 3.** Error rate (%) of the classifier using different appearance-based features.

	SIT	FD	PFD	NFD	AFD	SD
Basic features	<b>7</b>	18	27	31	21	32
Basic features+SIT	-	10	<b>9</b>	10	10	10

sequence length of the training samples. Table 3 shows how using concatenated feature vectors is not able to improve accuracy of the system here and simple SIT feature vectors are the most effective appearance-based features.

All former experiments use frames captured by the camera placed in front of the signer. We concatenate the weighted feature vectors of the front and lateral camera. Figure 4 shows the error rate of the classifier using minimum and average sequence length, with respect to the weights of the cameras. The minimum error rate occurs when the feature weight of the lateral camera is set to zero, which means that their frames are ignored. The error rate grows with increasing weight of the lateral camera. This result is probably caused by the occlusion of the hands. The HMM classifier with length of the average sequence length of training samples, increasing the weight of lateral camera, achieves smaller error rate in some portion of the diagram.

About half of the remaining errors are due to visual singletons in the dataset, which cannot be classified correctly using the leaving one out approach. For example, all but one of the signs for POSS show a movement of the right hand from the shoulder towards the right side of the signer, while the remaining one shows a movement that is directed towards the center of the body of the signer. This utterance thus cannot be classified correctly without further training ma-



**Fig. 4.** Error rate of the system with respect to the weight of cameras.

terial that shows the same movement. This is one of the drawbacks of the small amount of training data available.

A direct comparison to results of other research groups is not possible here, because there are no results published on publicly available data and research groups working on sign language or gesture recognition use databases that were created within the group.

## 6 Conclusion

In this paper, appearance-based features are used to recognize ASL words. These features already work surprisingly well for sign language word recognition. Furthermore, our system gives good results without any segmentation or tracking of the hands, which increases the robustness of the algorithm and reduces the computational complexity. If we use a color camera and the skin color probability instead of a black/white camera and the skin intensity in feature extraction, this approach can be generalized for other applications with the signers dressed differently and more cluttered background.

The visualization of the HMM and the analysis of the results show that the classifier is sensitive to different pronunciations of the same word. Therefore, we want to make use of explicit pronunciation modeling in the future. Furthermore, we will use explicit modelling of the variability of the images to cope with geometric changes in the appearance-based features. Using invariant features with respect to position and scale and modelling of variability will be helpful to make this feature vectors more effective. It makes the classifier more robust with respect to the changes of camera configuration, too. Obviously, the recognition of isolated models is only first step in the direction of recognition of complete sentences. One of the main problems in this direction is the scarceness of available data. We used publicly available data for the first time and we hope that other research groups will use this database and publish their results. We will apply our methods on larger databases in the future.

## References

1. J. Triesch and C. von der Malsburg. A System for Person-Independent Hand Posture Recognition against Complex Backgrounds. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(12):1449–1453, December 2001.
2. H. Birk, T.B. Moeslund, and C.B. Madsen. Real-Time Recognition of Hand Alphabet Gestures Using Principal Component Analysis. In *10th Scandinavian Conference on Image Analysis*, Laenranta, Finland, June 1997.
3. S. Malassiotis, N. Aifanti, and M.G. Strintzis. A Gesture Recognition System Using 3D Data. In *Proceedings IEEE 1st International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 190–193, Padova, Italy, June 2002.
4. S.A. Mehdi and Y.N. Khan. Sign Language Recognition Using Sensor Gloves. In *Proceedings of the 9th International Conference on Neural Information Processing*, volume 5, pp. 2204–2206, Singapore, November 2002.

5. K. Abe, H. Saito, and S. Ozawa. Virtual 3-D Interface System via Hand Motion Recognition From Two Cameras. *IEEE Trans. Systems, Man, and Cybernetics*, 32(4):536–540, July 2002.
6. J.L. Hernandez-Rebollar, R.W. Lindeman, and N. Kyriakopoulos. A Multi-Class Pattern Recognition System for Practical Finger Spelling Translation. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, pp. 185–190, Pittsburgh, PA, October 2002.
7. Y. Nam and K. Wohn. Recognition of Space-Time Hand-Gestures Using Hidden Markov Model. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 51–58, Hong Kong, July 1996.
8. B. Bauer, H. Hienz, and K.F. Kraiss. Video-Based Continuous Sign Language Recognition Using Statistical Methods. In *Proceedings of the International Conference on Pattern Recognition*, pp. 463–466, Barcelona, Spain, September 2000.
9. T. Starner, J. Weaver, and A. Pentland. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, December 1998.
10. C. Vogler and D. Metaxas. Adapting Hidden Markov Models for ASL Recognition by Using Three-dimensional Computer Vision Methods. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 156–161. Orlando, FL, October 1997.
11. C. Neidle, J. Kegl, D. MacLaughlin, B. Bahan, and R.G. Lee. *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure*. MIT Press, Cambridge, MA, 2000.
12. L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, 77(2):267–296, February 1989.

# Robust Person-Independent Visual Sign Language Recognition

Jörg Zieren and Karl-Friedrich Kraiss

Chair of Technical Computer Science  
RWTH Aachen University, Germany  
[www.techinfo.rwth-aachen.de](http://www.techinfo.rwth-aachen.de)

**Abstract.** Sign language recognition constitutes a challenging field of research in computer vision. Common problems like overlap, ambiguities, and minimal pairs occur frequently and require robust algorithms for feature extraction and processing. We present a system that performs person-dependent recognition of 232 isolated signs with an accuracy of 99.3% in a controlled environment. Person-independent recognition rates reach 44.1% for 221 signs. An average performance of 87.8% is achieved for six signers in various uncontrolled indoor and outdoor environments, using a reduced vocabulary of 18 signs.

The system uses a background model to remove static areas from the input video on pixel level. In the tracking stage, multiple hypotheses are pursued in parallel to handle ambiguities and facilitate retrospective correction of errors. A winner hypothesis is found by applying high level knowledge of the human body, hand motion, and the signing process. Overlaps are resolved by template matching, exploiting temporally adjacent frames with no or less overlap. The extracted features are normalized for person-independence and robustness, and classified by Hidden Markov Models.

## 1 Introduction

An important research area in computer vision is the tracking of objects in image sequences. This is often combined with the computation of features that describe the observed scene. Applied to human hands, classification methods known from speech recognition can be used to recognize gestures. The recognition of sign languages [1–5] is technically a special case of gesture recognition. It allows deaf people to intuitively control interactive devices in their first language [6].

Gestures can be defined in such a way that common computer vision problems like overlap, ambiguities, or minimal pairs do not occur. Signs, however, may only be chosen from a well-defined vocabulary. For sign language recognition, it is therefore essential to devise algorithms that perform reliably even in the aforementioned problematic situations.

This work describes a sign language recognition system that combines several properties previously not reported for a single application:

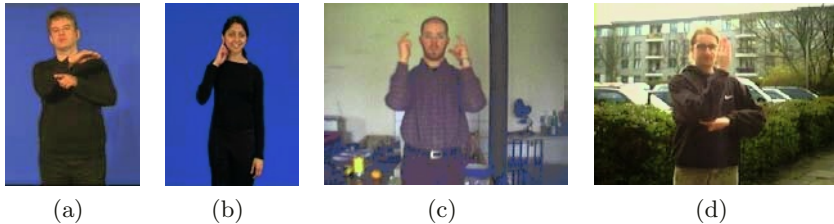
- It is non-intrusive, using a standard webcam ( $320 \times 240$  pixel, 25 fps) and a monocular frontal view. Requirements regarding video quality are very low.

- It operates with most uncontrolled backgrounds and lighting conditions, allowing mobile use.
- Person-independent operation is supported.
- Under ideal conditions, person-dependent recognition rates of 99.3% are achieved on a vocabulary of 232 signs from British Sign Language (BSL).

Section 2 presents the sign language video clips used for training and testing. System design and algorithms are described in section 3. Results for various recognition tasks can be found in section 4. Section 5 gives a short conclusion.

## 2 Application Scenario

A data base of BSL video clips (isolated signs, 2–3 seconds each) was created featuring two different recording setups. An average of 229 signs were performed by four signers, using strong lighting and homogenous backgrounds (see Fig. 1a,b). These form the system’s vocabulary and serve for both training and testing. A subset of these signs was recorded under real-world conditions with a regular webcam using six other persons (see Fig. 1c,d), and used only for testing. All signs were repeated five times. Since this work focuses on manual features, signs differing solely in non-manual features were not included in the vocabulary.



**Fig. 1.** Example frames from the training/test data base. a,b: Ideal conditions (384×288 pixel, 25 fps). c,d: Real-world conditions (320×240 pixel, 25 fps).

## 3 System Design

The recognition system can be divided into a feature extraction stage and a feature processing stage, each containing several modules, as shown in Fig. 2. Section 3.1 explains the feature extraction stage and the high level knowledge applied therein. The feature processing stage is discussed in section 3.2.

### 3.1 Feature Extraction

The following sections 3.1.1 to 3.1.4 describe the four feature extraction modules.

**3.1.1 Face Detection and Threshold Segmentation.** A person and illumination independent skin color model [7] is used to create a skin probability map that allows robust detection of the signer’s face and hands, but produces



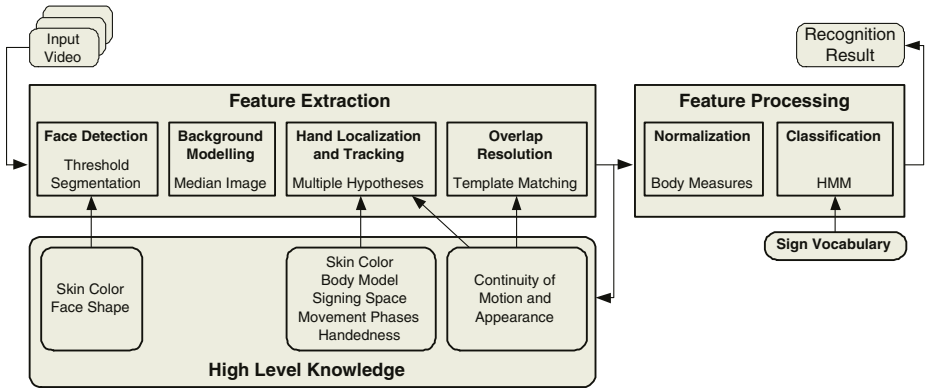


Fig. 2. Schematic of the recognition system.



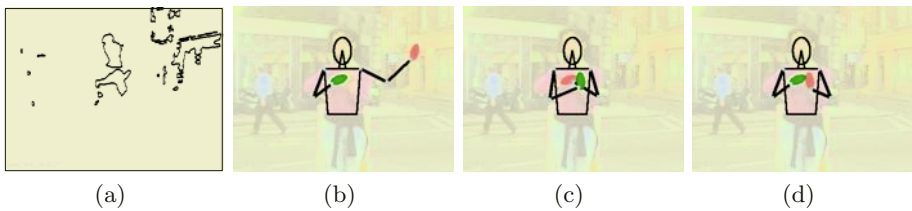
Fig. 3. a: Background model. b: Example frame. c: Foreground pixels (white).

numerous false alarms in real-world settings. Background modelling as described in section 3.1.2 cannot be applied here because the face itself is mostly static. The skin probability threshold for the following segmentation is found automatically. A metric has been defined to quantify a given boundary’s deviation from that of an average face in terms of several geometric features (position, size, orientation, axis ratio, compactness). A number of thresholds is then tested and the one which yields the face candidate blob with the lowest deviation is chosen.

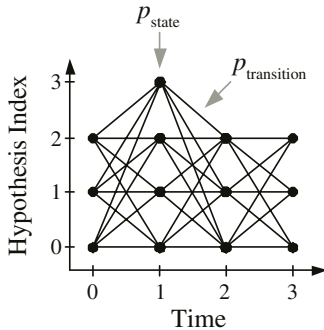
The use of skin color leads to the common restriction that the signer must wear long-sleeved, non-skin-colored clothing to allow a color-based segmentation of face and hands at least in the absence of overlap [1, 2, 8, 9].

**3.1.2 Background Modelling.** After the face has been detected and segmented, the image background is excluded from further processing to reduce computational cost and the number of distractors. This is done on pixel level since a semantic interpretation is not available at this stage. A simple yet effective method to create a background model  $p_b(x_0, y_0) = (r_b, g_b, b_b)$  for coordinates  $(x_0, y_0)$  is to compute the median of all pixels  $p(x_0, y_0, t)$  over time  $t$ . Fig. 3 shows the background model for a complete clip (a) and its application to an individual frame (b,c). In comparison, approaches that model  $p_b(x_0, y_0)$  as a mixture of Gaussians [10–12] proved less robust on short video clips and require multiple parameters to be specified, whereas the median is parameter-free.

**3.1.3 Hand Localization and Tracking.** The only image cue used for hand localization is color. This is motivated by the hands' extremely variable appearance, which prevents the use of shape or texture cues. Especially at typical image resolutions around  $320 \times 240$ , these cannot be exploited reliably. The principal drawback of the color cue is its susceptibility to false alarms. It is therefore important to devise tracking algorithms that explicitly deal with ambiguities. Fig. 4a shows the skin color segmentation of a typical scene (Fig. 3b). This observation does not allow a direct conclusion as to the actual hand configuration. Instead, there are multiple interpretations, or hypotheses, as visualized in Fig. 4b–d. Previous observations may suggest a certain interpretation, but they may be incorrect, so no decision should be made at this stage.



**Fig. 4.** a: Skin color segmentation. b–d: Subset of hypotheses to a (correct: d).



**Fig. 5.** Hypothesis space and probabilities for states and transitions.

Therefore, the tracking stage creates all conceivable hypotheses for every frame. Transitions are possible from each hypothesis at time  $t$  to all hypotheses at time  $t + 1$ , resulting in a state space as shown exemplarily in Fig. 5. The total number of paths through this state space equals  $\prod_t N(t)$ , where  $N(t)$  denotes the number of hypotheses at time  $t$ . Provided that the skin color segmentation detected both hands and the face in every frame, one of these paths represents the correct tracking result. In order to find this path (or one as close as possible to it), probabilities are computed heuristically that indicate the likeliness of each hypothesized configuration,  $p_{state}$ , and the likeliness of each transition,  $p_{transition}$  (see Fig. 5). High level knowledge is applied as follows:

- A biomechanical body model is created from the face position, face size, and hands position. Configurations that are physiologically unlikely or do not occur in sign language reduce  $p_{\text{state}}$ . This considers the three phases of a sign (preparation, stroke, retraction), as well as the signer’s handedness (which has to be known in order to correctly interpret the feature vector).
- Even in fast motion, the area enclosed by the hand’s boundary changes only slowly between successive frames at 25 fps. In case of the start or cessation of an overlap, the area drops or rises by the size of the individual blobs. Thus, at time  $t$ , an estimate can be computed for time  $t + 1$ . With increasing deviation of the actual from the expected area,  $p_{\text{transition}}$  is reduced.
- Similarly, hand position changes slowly so that coordinates at time  $t$  may serve as a prediction for time  $t + 1$ . Kalman filters have been found not to increase tracking performance since the direction of movement varies too quickly during the stroke phase. Also, they would prohibit the application of the Viterbi algorithm (see below) by adding a memory to each path.
- The above criteria tend to favor slowly moving distractors. To counter this effect, the average color difference of a hand blob’s pixels between the current and the previous frame is computed. Higher values increase  $p_{\text{state}}$ .

To search the hypothesis space, the Viterbi algorithm [13] is applied in conjunction with pruning of unlikely paths.

The multiple hypotheses tracking approach ensures that all available information is evaluated before the final tracking result is determined. The tracking stage can thus exploit, at time  $t$ , information that becomes available only at time  $t_1 > t$ . Errors are corrected retrospectively as soon as they become apparent.

**3.1.4 Overlap Resolution.** When two or more objects overlap each other in the image, the skin color segmentation yields only a single blob for multiple objects. This happens frequently in sign language. A direct extraction of meaningful features is not possible in this case. Low contrast, low resolution, and the hands’ variable appearance do not allow a separation of the overlapping objects by an edge-based segmentation either. Most of the geometric features available for unoverlapped objects can therefore not be computed for overlapping objects and are interpolated linearly. However, a hand’s appearance is sufficiently constant over several frames for template matching to be applied. Using the last unoverlapped view of each overlapping object as a template, at least position features – which fortunately carry much information – can be reliably computed during overlap. The accuracy of this method decreases with increasing template age, but the multiple hypotheses framework allows to also access the first unoverlapped view after the cessation of an overlap and use it as a second template. The system prefers whichever template produced the better match. This effectively halves the maximum template age and increases precision considerably.

### 3.2 Feature Processing

The geometric features computed by the tracking stage to describe each hand’s configuration are:

- Coordinates  $x, y$  of the center of gravity (COG), and their derivatives  $\dot{x}, \dot{y}$
- Area  $a$ , and its derivative  $\dot{a}$
- Ratio  $r$  of inertia parallel and orthogonal to the main axis
- $\sin 2\alpha$  and  $\cos \alpha$  of the main axis orientation  $\alpha$
- Compactness  $c$  and eccentricity  $e$  [14]

These elements constitute the 22-dimensional feature vector forwarded to the classifier and used in the application of high level knowledge. Position ( $x_F, y_F$ ) and width ( $w_F$ ) of the face are used for normalization (see below), but are not included in the feature vector. Sections 3.2.1 and 3.2.2 explain the two modules that constitute the feature processing stage as shown in Fig. 2.

**3.2.1 Normalization.**  $a$  depends on image resolution as well as on the signer’s distance to the camera.  $x$  and  $y$  additionally depend on the signer’s position in the image. For a person-independent real-world application, these features have to be normalized. The feature processing stage estimates the position of the signer’s shoulders from  $x_F, y_F$ , and  $w_F$ , and specifies the position of the left/right hand relative to the left/right shoulder. Distances and areas are normalized by  $w_F$  and  $w_F^2$ .

**3.2.2 Classification.** After cropping idle feature vectors at the beginning and the end, and an optional mirroring for left-handed signers, the feature vector sequence is forwarded to the HMM classifier. The system allows to only activate a subset of all HMMs, depending on the application context. For use in an interactive dialog, this would be the items in the current menu.

## 4 Evaluation

Due to the lack of standardized benchmarks, recognition rates of different systems cannot be compared directly since they are valid only for the actual test scenario. Nevertheless, they give a general idea of a system’s performance and provide a useful measure when parameters are varied.

Tab. 1 shows the person-dependent recognition rates from a leaving-one-out test for the four signers recorded as shown in Fig. 1a,b and various test video

**Table 1.** Person-dependent recognition rates in controlled environments.

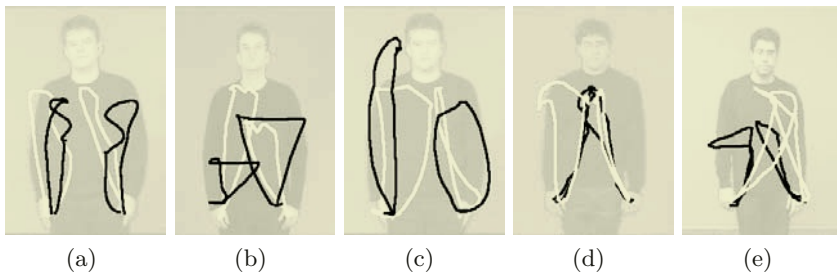
Test Video Resolution	Features	Signer, Vocabulary Size				
		Ben 235 signs	Michael 232 signs	Paula 219 signs	Sanchu 230 signs	$\emptyset$ 229 signs
$384 \times 288$	all	98.7%	99.3%	98.5%	99.1%	98.9%
$192 \times 144$	all	98.5%	97.4%	98.5%	99.1%	98.4%
$128 \times 96$	all	97.7%	96.5%	98.3%	98.6%	97.8%
$96 \times 72$	all	93.1%	93.7%	97.1%	95.9%	94.1%
$384 \times 288$	$x, \dot{x}, y, \dot{y}$	93.8%	93.9%	95.5%	96.1%	94.8%

resolutions. The training resolution was always  $384 \times 288$ . Vocabulary size is specified for each signer since the number of recorded signs varies slightly. Interestingly, COG coordinates alone allow recognition rates up to 96% for 230 signs. On a 2 GHz PC, processing took an average of 11.79s/4.15s/3.08s/2.92s per sign, depending on resolution. Low resolutions cause only a slight decrease in recognition rate but reduce processing time considerably. Compared to previous results, an increase in both vocabulary size and recognition rate has been achieved. Higher performance has only been reported for intrusive systems.

**Table 2.** Person-independent recognition rates in controlled environments.

Training Signer(s)	Test Signer	Vocabulary Size	n-Best Rate		
			1	5	10
Michael	Sanchu	205	37.1%	58.1%	65.0%
Paula, Sanchu	Michael	218	31.2%	54.7%	63.4%
Ben, Paula, Sanchu	Michael	224	32.9%	57.8%	67.1%
Ben, Michael, Paula	Sanchu	221	44.1%	69.6%	79.1%
Ben, Michael, Sanchu	Paula	212	31.5%	57.8%	68.5%
Michael, Sanchu	Ben	206	3.7%	11.7%	15.3%

Tab. 2 shows results for person-independent recognition. Since the signers used different signs for some words, the vocabulary has been chosen as the intersection of the test signs with the union of all training signs. In case of multiple training signers, some signs (around 5%) were therefore only trained with a subset of the training signers. No selection has been performed otherwise, and no minimal pairs have been removed. As expected, performance drops significantly. This is caused by strong interpersonal variance in signing. In particular, Ben’s signing differs from the other three. Fig. 6 shows COG traces for identical signs done by different signers to visualize the degree of deviation. Recognition rates are also affected by the exact constellation of training/test signers and do not necessarily increase with the number of training signers.



**Fig. 6.** Interpersonal variance. Traces from Michael (white) and Paula (black) signing “autumn” (a), “recruitment” (b), “tennis” (c), and Michael (white) and Ben (black) signing “distance” (d), “takeover” (e).

**Table 3.** Person-independent recognition rates in uncontrolled environments.

Vocabulary Size	Test Signer						
	Christian	Claudia	Holger	Jörg	Markus	Ulrich	∅
6	96.7%	83.3%	96.7%	100%	100%	93.3%	95.0%
18	90.0%	70.0%	90.0%	93.3%	96.7%	86.7%	87.8%

Person-independent performance in uncontrolled environments is difficult to measure since it depends on multiple parameters (signer, vocabulary, background, lighting, camera). Tab. 3 shows results for small vocabularies. Each person was recorded in a different environment (see Fig. 1c,d). The classifier was trained with Ben, Michael, and Paula. The feature extraction stage performed well in most scenarios, but inter-personal variance does not allow to recognize larger vocabularies with comparable accuracy. This problem is aggravated by noise and outliers invariably introduced in the features when operating in real-world settings.

## 5 Conclusion

High recognition performance has been achieved for person-dependent classification. The presented system is also suitable for person-independent real-world applications where small vocabularies suffice, such as controlling interactive devices. Two main challenges can be identified for robust person-independent recognition of larger vocabularies: Accurate feature extraction in real-world conditions, and handling inter-personal variance in feature processing. We are confident that multiple hypothesis tracking solves the former, while the latter will clearly be subject of further research.

## References

1. Starner, T., Weaver, J., Pentland, A.: Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. In: IEEE PAMI. (1998)
2. Tanibata, N., Shimada, N., Shirai, Y.: Extraction of Hand Features for Recognition of Sign Language Words. In: Proc. Int. Conf. Vision Interface. (2002)
3. Bauer, B., Kraiss, K.F.: Video-Based Sign Recognition using Self-Organizing Subunits. In: Lecture Notes in Artificial Intelligence 2298. (2002)
4. Zieren, J., Kraiss, K.F.: Non-Intrusive Sign Language Recognition for Human-Computer Interaction. In: Proceedings of the IFAC-HMS Symposium. (2004)
5. Yang, M.H., Ahuja, N., Tabb, M.: Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition. In: IEEE TPAMI. Volume 24. (2002)
6. Akyol, S., Canzler, U.: An Information Terminal using Vision Based Sign Language Recognition. In Büker, U., Eikerling, H.J., Müller, W., eds.: ITEA Workshop on Virtual Home Environments, VHE Middleware Consortium. Volume 12. (2002)
7. Jones, M., Rehg, J.: Statistical Color Models with Application to Skin Detection. Technical Report CRL 98/11, Compaq Cambridge Research Lab (1998)

8. Imagawa, K., Lu, S., Igi, S.: Color-Based Hand Tracking System for Sign Language Recognition. In: IEEE Int. Conf. on Autom. Face and Gesture Recognition. (1998)
9. Huang, C.L., Huang, W.Y.: Sign language recognition using model-based tracking and a 3D Hopfield neural network. In: Machine Vision and Applications. Volume 10. (1998)
10. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Computer Vision and Pattern Recognition 1999. Volume 2. (1999)
11. KaewTraKulPong, P., Bowden, R.: An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection. In: AVBS. (2001)
12. Porikli, F., Tuzel, O.: Human Body Tracking by Adaptive Background Models and Mean-Shift Analysis. Technical report, Mitsubishi Electric Research Lab. (2003)
13. Rabiner, L., Juang, B.H.: An Introduction to Hidden Markov Models. IEEE ASSP Magazine **3** (1986)
14. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. Brooks Cole (1998)

# A 3D Dynamic Model of Human Actions for Probabilistic Image Tracking

Ignasi Rius, Daniel Rowe, Jordi Gonzàlez, and Xavier Roca

Centre de Visió per Computador/Department of Computer Science  
Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain  
`irius@cvc.uab.es`

**Abstract.** In this paper we present a method suitable to be used for human tracking as a *temporal prior* in a particle filtering framework such as CONDENSATION [5]. This method is for predicting feasible human postures given a reduced set of previous postures and will drastically reduce the number of particles needed to track a generic high-articulated object. Given a sequence of preceding postures, this example-driven transition model probabilistically matches the most likely postures from a database of human actions. Each action of the database is defined within a PCA-like space called *UaSpace* suitable to perform the probabilistic match when searching for similar sequences. So different, but feasible postures of the database become the new predicted poses.

## 1 Introduction

The analysis of motion in image sequences involving humans has become a great interest area in computer vision because of the wide amount of promising applications it brings, i.e. automatic surveillance, sports performance analysis, advanced interfaces, augmented reality and motion synthesis among others. This challenging domain is referred as *Human Sequence Evaluation* (HSE) in the framework presented by González in [3], and provides a general scheme for producing useful human motion descriptions from images suitable to be used for such applications.

The HSE framework divides the task of evaluating sequences of images involving human motion in several layers or modules, each one encapsulating different domains of knowledge. Hence, the interpretation of human motion is treated as a transformation process from level to level. We focus on the transformation process between the 3D human body configurations from 2D image sequences. This tracking and reconstruction task of articulated 3D human motion is a key point of HSE and has become a wide research topic in the last years [8].

Among others, one critical issue is the high dimensionality and the non-linearity of the articulated rigid objects to be tracked. For instance, if we consider a 3D body model of 12 joints with 3 Degrees of Freedom (DOF) per joint, it results in a model with 36 DOF, which means that our tracking algorithm must estimate at least 36 parameters at each time step. So several optimization techniques are usually applied.



The remainder of this paper is organized as follows. Section 2 explains the probabilistic framework used to face the tracking problem. Section 3 describes the human action model employed in this work. Section 4 focuses on the problem of the probabilistic search within the space of actions. Section 5 shows some experimental results, and section 6 concludes this paper.

## 2 Probabilistic Tracking Framework

The objective of visual tracking is to estimate the parameters of our model  $\phi_t$  at time  $t$  given the sequence of images  $\mathbf{I}_t$  up to that moment. In other words, we need to compute the posterior *probability density function* (pdf)  $p(\phi_t|\mathbf{I}_t)$  over the parameters  $\phi_t$  of the model to be tracked at time  $t$ . Thus, using the Bayes' rule, we formulate the computation of our model parameters over time as [2]:

$$p(\phi_t|\mathbf{I}_t) = k p(I_t|\phi_t) \int p(\phi_t|\phi_{t-1}) p(\phi_{t-1}|\mathbf{I}_{t-1}) dt, \quad (1)$$

where  $\phi_t$  represents the pose of the human body at time  $t$ ,  $\mathbf{I}_t$  is the image sequence up to time  $t$ ,  $k$  is a normalizing factor,  $p(I_t|\phi_t)$  is the *likelihood* of observing the image  $I_t$  given the parametrization  $\phi_t$  of our model at time  $t$ , and finally  $p(\phi_t|\phi_{t-1})$  is the *temporal prior*, or dynamic model in this work.

The recursive Bayesian filter provides the theoretical optimal solution. It decomposes the problem in two differentiated steps, i.e. *prediction* and *update*. On the prediction step, a dynamic model is used to derive the prior pdf at time  $t$  from the already computed posterior pdf at time  $t-1$ . On the update step, the *likelihood* function is used to compute the posterior pdf at time  $t$ .

Unfortunately, Eq.(1) relies on an integral which cannot be analytically calculated unless strong assumptions about Gaussianity and linearity on the involved distributions are made. Instead, we can approximate the true posterior distribution  $p(\phi_t|\mathbf{I}_t)$  by means of a *particle filter* [1, 5]. Particle filtering is based on Monte Carlo Simulation, thus, our posterior distribution at time  $t$  is represented by a set of samples or particles that in our case define a particular human body posture. Each particle has its own probability of being propagated over time depending on how likely is its corresponding body posture to be found on the image  $I_t$ . If a particle is selected to be propagated at time  $t$ , a transition model or *dynamic model* is used to predict the new location in the parameter space at time  $t+1$ , i.e. the new particle at the following time step.

This Bayesian model-based tracking approach brings us a principled way for considering multiple hypotheses about the human body posture, and allows us to integrate prior knowledge about the non-linear human dynamics into the tracking making it more robust and efficient.

Since the dimensionality of the parameters space is very large in 3D human motion tracking, a large number of particles may be needed to successfully track our model parameters over time. However, the number of particles grow exponentially with the model dimensionality [6]. To overcome this, we need an appropriate dynamic model in order to reduce the number of particles needed

to make the tracking task possible. This *temporal prior* should capture the behaviour of human motion accurate enough to predict only new feasible postures, but generic enough to be able to track any actor and any human motion.

The aim of this work is to present a temporal prior derived from [7], which is suitable to be used by the particle filter. Hence, the proposed model will propagate the parameters of our human body model over time while reducing the number of particles required to track a 3D human body model during a performance. The goal is focused in generating only the most plausible body postures within the performance of a particular action, rather than attempting to randomly propagate the parameters of a generic, high-articulated object.

### 3 Human Action Modeling Using *p-actions*

Our method learns the implicit probabilistic model of 3D human motion by using an example-based approach. Our dynamic model will use a database of learnt actions in order to predict the most suitable future body poses given a reduced set of the history of estimated poses. We perform a probabilistic search within a PCA-like space, called *UaSpace* [3], which is built from a training set of human motions acquired with a commercial Motion Capture system.

In this work we use the human action model and the human action space defined in [4], called *p-action* and *aSpace* respectively. We show how to employ this action model to develop a dynamic model suitable to be used for human posture prediction which focuses and restricts the search space to those postures with highest likelihood values in factored sampling techniques.

An action will be represented as a sequence of postures, so a proper body model is required, which is learnt from examples. The training data has been acquired using a commercial Motion Capture system. A set of 19 reflective markers were placed on several characteristic points of the subject's body. The body model employed is composed of twelve rigid body parts (hip, torso, shoulder, neck, two thighs, two legs, two arms and two forearms) and fifteen joints. These joints are structured in a hierarchical manner, where the root is located at the hip. We represent the human body by 37 parameters which describe the relative elevation and orientation of each limb which are natural to be used for limb movement description. See [3, 4] for further details.

As a result, the training data set for each action  $\mathbf{A}_i$  is composed of  $r_i$  sequences  $\mathbf{A}_i = \{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_{r_i}\}$ , each one corresponding to a cycle or a performance of the action to be modeled.

Thus, we define the complete set of human postures for an action  $\mathbf{A}_i$  as:

$$\mathbf{A}_i = \{x_1, x_2, \dots, x_{f_i}\}, \quad (2)$$

where each  $x_j$  of dimensionality  $n \times 1$  stands for the 37 values of the human body model described previously and  $f_i$  refers to the overall number of training postures for this particular action  $\mathbf{A}_i$ .

Then, we perform a Principal Component Analysis (PCA) on the training set  $\mathbf{A}_i$ , and compute its *aSpace* as defined in [4]. Afterwards, for each performance

$\mathbf{H}_j$ , we consider its projections within the *aSpace* of the captured postures as the control values for an interpolating curve  $\mathbf{g}_j(p)$ , which is computed using a standard cubic-spline interpolation algorithm. The parameter  $p$  refers to the temporal variation of the posture, which is normalized for each performance, that is,  $p \in [0, 1]$ . This process is repeated for each performance of the learning set, and a mean manifold  $\mathbf{g}(p)$  is obtained by interpolating between the means of  $\mathbf{g}_j(p)$  for each index  $p$ .

After, a key-frame set  $\mathbf{K}$  is found for each action by using the Mahalanobis distance, and the final human action model is represented as a parametric manifold  $\mathbf{f}(p)$ , called *p-action*, which is built by interpolation between those key-frames.

For our purposes, we need a common space where all the *p-actions* can be represented. We denote this space as the Universal *aSpace* or *UaSpace*, and is defined in the same fashion as the single *aSpace* for each action, but using all the postures from all the performances from all the actions of our database. After applying PCA, the first  $b^U = 15$  eigenvectors are chosen to determine the 95% of the variance, and will constitute the basis of the space  $\Omega^U$  where all the *p-actions* will be represented.

Finally, an action  $\mathbf{A}_i$  is modeled within the *UaSpace* as:

$$\Gamma^{A_i} = (\Omega^U, \mathbf{K}^{A_i}, \mathbf{f}^{A_i}), \quad (3)$$

where  $\Omega^U$  defines the eigenvectors and the eigenvalues of the *UaSpace*, and  $\mathbf{K}^{A_i}$ ,  $\mathbf{f}^{A_i}$  correspond to the key-frames and the parametric manifold that defines the *p-action*, respectively.

Closer points between different manifolds correspond to similar human postures of several actions. In fact, the distance between two points in the *UaSpace* can be considered as a measure of similarity between human postures.

## 4 Probabilistic Dynamic Model

Multiple hypotheses can be generated by considering different dynamical models. We consider the human action model  $\Gamma^{A_i}$  defined before as the basis for those dynamical models which can help to generate new samples over time within a probabilistic framework. As postures can be shared among different actions (such as in sitting, squatting and tumbling, for example), we need a probabilistic model which can deal with multiple hypotheses while predicting new postures. Fortunately, the *UaSpace* provides the framework where multiple motion models can be learnt and recognized.

The goal of a dynamic model is to predict new body postures  $\phi_{t+1}$  at time  $t + 1$  given the history of the observed motion  $\Phi_t$  from time  $t - d$  to time  $t$ . In our approach, the motion database used to build the dynamic model is derived from all the *p-actions* represented within the *UaSpace* described in the previous section. In order to obtain a set of body postures from each parametric manifold, each cubic-spline  $\mathbf{f}^{A_i}(p)$  is sampled at a constant rate considering that  $p \in [0, 1]$ .

We denote each projected human posture of dimension  $b^U$  within the *UaS-space* as  $\psi_i$ , and  $\Psi_i = [\psi_i^T, \dots, \psi_{i-d}^T]^T$  refers to the  $(d \times b^U)$ -dimensional vector containing all the postures in the database from location  $i - d$  to location  $i$ , i.e. the history of motion of the last  $d$  postures. In a similar fashion, let  $\phi_t$  be the estimated posture at time  $t$  in the tracking framework described in section 2, and  $\Phi_t = [\phi_t^T, \dots, \phi_{t-d}^T]^T$  the estimated sequence from time  $t - d$  to time  $t$ .

To perform the probabilistic tracking using the particle filtering approach, our final goal is to generate new particles at the prediction step, i.e. to draw samples  $\phi_t^s$  from the dynamic model  $p(\phi_t|\Phi_{t-1})$ . Following the approach described by Sidenbladh in [7], we can rewrite this distribution as:

$$p(\phi_t|\Phi_{t-1}) = p(\phi_t|\Psi_{i-1})p(\Psi_{i-1}|\Phi_{t-1}), \quad (4)$$

where  $p(\phi_t|\Psi_{i-1})$  is defined as 1 if  $\phi_t = \psi_i$ , or 0 otherwise.

Thus, sampling from the prior  $p(\phi_t|\Phi_{t-1})$  corresponds to sampling from the distribution  $p(\Psi_{i-1}|\Phi_{t-1})$ . This can be seen as performing a probabilistic search of the estimated motion  $\Phi_t$  with a stored sequence  $\Psi_i$  from the database. Assuming that sequences of estimated postures follow a Gaussian distribution around matching sequences on the database, i.e.:

$$\Psi_i = \Phi_t + \eta(\Delta_d), \quad (5)$$

the matching probability is given by

$$p(\Psi_i|\Phi_t) = k e^{-\frac{1}{2}(\Psi_i - \Phi_t)^T \Delta_d^{-1}(\Psi_i - \Phi_t)}, \quad (6)$$

where  $k$  is a normalizing factor.

The covariance matrix  $\Delta_d$  is defined by calculating the covariance  $\Delta$  of all the postures  $\psi_i$  from the database, and storing  $d$  copies of  $\Delta$  along the diagonal of the  $d \cdot b^U \times d \cdot b^U$  covariance matrix  $\Delta_d$ . By doing this, we give the same importance to each posture when matching the sequences, see [7] for details.

Thus, the dynamic model will estimate feasible human postures for tracking by searching only for the most likely stored postures from the database, and adding an empirically determined Gaussian noise term to them. Since this is a probabilistic model, we can generate  $n$  new different particles  $\phi_t^s$  at each time step by sampling  $n$  times from the distribution  $p(\phi_t|\Phi_{t-1})$  defined using the learnt *p-actions* from the database.

## 5 Experimental Results

The dynamic model has been trained with 9 different basic actions (*aRun*, *aWalk*, *aBend*, *aSit*, *aJump*, *aSkip*, *aSquat*, *aTumble* and *aKick*) considering near 100 postures for each action, by sampling the parametric manifolds  $\mathbf{f}^{A_i}(p)$  that represent each action  $A_i$  at a constant rate with a sampling step of 0.01,  $p \in [0, 1]$ .

The testing set consisted in 5 performances per action, each one performed by 9 different actors. This results in 45 performances of all the actions which were not included in the training set for the calculation of the *p-actions*.

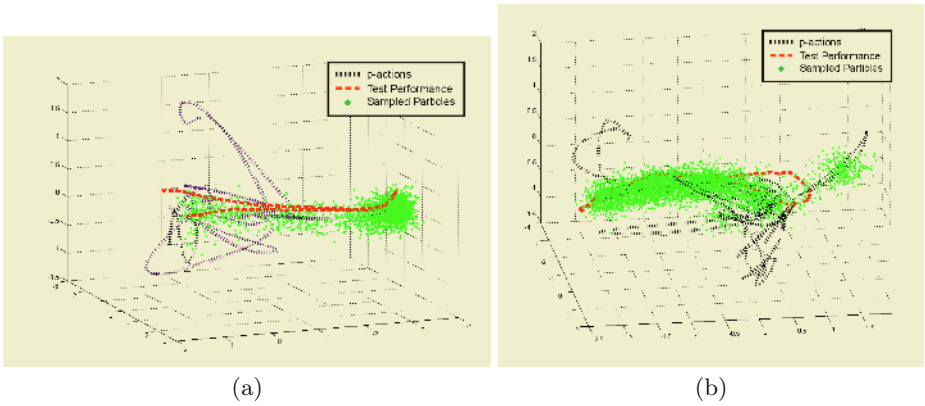
**Table 1.** Confusion Matrix in percentages.

Action	aRun	aWalk	aBend	aSit	aJump	aSkip	aSquat	aTumble	aKick
aRun	97	0	0	0	0	0	0	0	3
aWalk	0	72	0	1	1	23	1	1	1
aBend	0	9	83	2	3	0	1	1	1
aSit	0	21	3	65	0	4	4	3	0
aJump	8	3	0	1	70	7	1	1	8
aSkip	0	10	0	0	0	85	0	0	5
aSquat	5	0	4	0	1	0	90	0	0
aTumble	0	0	0	2	2	0	0	95	1
aKick	1	11	1	2	13	21	2	1	48

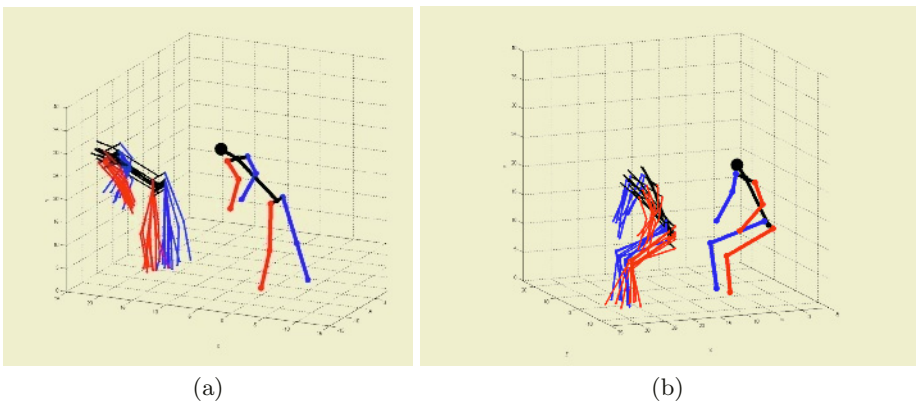
In order to explore the coverage of the search space performed by our dynamic model, we generated all the possible motion histories of length  $d$  ( $d = 10$ ) for each test performance, and sampled 100 new postures or particles per each motion history following the procedure described above. After doing this for all the test performances, the confusion matrix shown in Table 1 was generated, where each row indicates the class, or  $p$ -action of the tested subsequence, and each column corresponds to the class of the sampled particle using a minimum Mahalanobis distance criteria.

This table shows that our predictions are not too focused on an specific action, but still cover the truly performed action well enough. These results reflect the fact that some actions share a lot of similar postures between each other, especially at the beginning and at the end of the performances. This situation is very well handled by our dynamic model, since it is able to throw multiple hypotheses when the given subsequence is very similar in several actions, so we do not restrict the searching space to any of them. These hypotheses will be propagated over time by the particle filter until some of them become very unlikely over time. For instance, looking at Table 1, we observe that the action of *aWalk* has a lot of similarities with the action of *aSkip*. In the *aSkip* action a subject starts walking, and after some frames it passes over some obstacle. Thus, the two actions share a lot of postures, especially at the beginning and at the end. Therefore, multiple hypotheses on what is the agent doing must be thrown on that situations, which is fulfilled by our dynamic model. We can find a similar situation between the *aBend* and the *aWalk* actions, and between the *aJump*, *aRun* and *aKick*. The table also shows that most of the actions only share a few postures, or none at all. So, this result is useful for establishing relationships between the involved actions. Further study needs to be done in order to determine similarities between parts of the same action, and not the action as a whole, in order to analyse the predictions made by the dynamic model.

In Fig 1.(a) the first 3 dimensions of the *UaSpace* are drawn together with a *aBend* test performance (dashed line). We have generated particles up to the middle of the performance by our dynamic model and plotted them on the *UaSpace* as single dots. We can observe that the predictions made at the



**Fig. 1.** Sampling from the dynamic model within the *UaSpace*. See text for details.



**Fig. 2.** Predicted human postures for the *aBend* and *aSit* actions. See text for details.

beginning of the action are split mainly between the bending and other actions such as *aWalk*, *aJump* and *aSit*. But, as the performance goes over time, almost all the predictions are concentrated along the bending *p-action*, since it becomes very different to the other actions. A similar situation for a *aSit* test performance is shown in Fig.1(b).

In Fig 2.(a) and 2.(b) we show the true posture on the right and a set of predicted postures on the left for a particular frame of the same *aBend* and *aSit* performances used in Fig 1. The set shown is randomly selected from the 100 predicted postures. The results obtained point out that this dynamic model is focused on generating the most suitable postures while performing an action, and naturally reduces the searching space avoiding the evaluation of improbable and impossible body configurations.

## 6 Conclusions and Future Work

This paper presents a temporal prior distribution suitable to be used as a dynamic human body model for tracking. The drawn particles from this distribution correspond to predicted feasible poses of the body given the history of estimated poses over time. The method learns a human motion model from a database of 3D actions acquired with a commercial Motion Capture System.

The results point out that this procedure, if used in a particle filtering framework, will drastically reduce the number of particles needed to track a human body while performing an action. Even though the proposed example-based dynamic model is less flexible than generic models for articulated objects motion, it is generic and accurate enough for making the tracking of human motion an achievable task.

Future research relies on integrating this approach into a particle filtering framework and developing appropriate likelihood measures for human bodies in 2D images. To reduce the problems of extrapolating from the *p-action* model, a more refined action model could be developed by probabilistically modeling each action using Mixtures of Gaussians, for example. Furthermore, transitions between actions could be naturally modelled by interpolating between the key-frames of several *p-actions*. Another open issue is the high computational cost of the probabilistic search, which could be addressed by efficient indexing the motion database.

## Acknowledgments

This work has been supported by the Spanish CICYT TIC2003-08865.

## References

1. M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.
2. Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and data association*. Academic Press, 1988.
3. Jordi González. *Human Sequence Evaluation: the Key-frame Approach*. PhD thesis, Universitat Autònoma de Barcelona, May 2004.
4. Jordi González, Javier Varona, F. Xavier Roca, and J. José Villanueva. Analysis of human walking based on aSpaces. *3rd International Workshop on Articulated Motion and Deformable Objects (AMDO'2004)*, September 2004.
5. M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
6. O. King and David A. Forsyth. How does CONDENSATION behave with a finite number of samples? In *ECCV (1)*, pages 695–709, 2000.
7. Hedvig Sidenbladh, Michael J. Black, and Leonid Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV (1)*, pages 784–800, 2002.
8. L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36:585–601, 2003.

# Extracting Motion Features for Visual Human Activity Representation\*

Filiberto Pla<sup>1</sup>, Pedro Ribeiro<sup>2</sup>, José Santos-Victor<sup>2</sup>, and Alexandre Bernardino<sup>2</sup>

<sup>1</sup> Computer Vision Group, Departament de Llenguatges i Sistemes Informàtics  
Universitat Jaume I, 12071 Castellón, Spain  
Filiberto.Pla@lsi.uji.es

<sup>2</sup> Computer Vision Lab – VisLab, Instituto de Sistemas e Robótica  
Instituto Superior Técnico, Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal  
{pribeiro,jasv,alex}@isr.ist.utl.pt

**Abstract.** This paper presents a technique to characterize human actions in visual surveillance scenarios in order to describe, in a qualitative way, basic human movements in general imaging conditions. The representation proposed is based on focus of attention concepts, as part of an active tracking process to describe target movements. The introduced representation, named “focus of attention” representation, FOA, is based on motion information. A segmentation method is also presented to group the FOA in uniform temporal segments. The segmentation will allow providing a higher level description of human actions, by means of further classifying each segment in different types of basic movements.

## 1 Introduction

Monitorizing human activity is one of the most important visual tasks to be carried out in visual surveillance scenarios. This task includes processes like target tracking, human activity characterization and recognition, etc. Human activity characterization and recognition is a special topic that has been addressed in the literature from different points of views and for different purposes [8] [10] [2] [1] [9].

In the work described here, the objective was to characterize, aimed at building a feature representation for further recognition, the human activity of people in typical visual surveillance scenarios, like airport lounges, public building halls, commercial centers, etc., with a great variety of human action types and ordinary, rather poor, imaging conditions. The main idea of the proposed techniques is to perform a general description of basic human movements, extracting some visual cues that can help to understand the people’s actions in higher level recognition tasks.

In order to understand the activity of a person in a given scenario, the human movement can be described as a composition of two different types of movements:

1. The movements that a person performs with respect to the environment, that is, the analysis of trajectories and dynamics, performing target tracking. Some of the works are based only on this information [3].

---

\* Work partially supported by grant from the *Spanish Ministry of Science and Education* PR2004-0333, and CAVIAR IST-2001-37540 project from European Union.



2. The movements that the different parts of the body a person performs during a certain action, with respect to the body point of view.

According to the classification described by [2], human activity recognition approaches can be divided in three different groups. First group are *Generic model recovery* approaches, in which, at each time, the person pose is recovered trying to fit it with a 3D body model. These approaches strongly depend on an accurate 3D feature extraction from the image, which usually needs human intervention and controlled environments to facilitate image measurements [6].

*Appearance-based models* are an alternative to 3D model recovery, appearance based models rely on 2D information extracted from the images, either raw grey level distributions or other processed image features, like region templates, contours, etc. where an action is described as a sequence of 2D poses of the moving target [7].

Finally, *motion-based recognition* techniques try to recognize the human activity by analyzing directly the motion itself, without referring it to any static model of the body. The rationale of these approaches lie in the fact that different movements of the body produce defined motion patterns in the image domain [2] [1] [5] [9]. Therefore, some of these works use optical flow measurements as motion features to recognize human activities [10] [8].

The approach presented here is included in the motion-based recognition techniques, aiming at characterizing human activities directly from the motion information. In particular, we will use optical flow information, focusing our attention to the movements of different parts of the body, trying to characterize basic body movements. Therefore, we will assume that a certain target extraction and tracking has already been performed, that is, we will keep our “active” attention to the target only, centering our target in our field of view, the fovea, for further analysis.

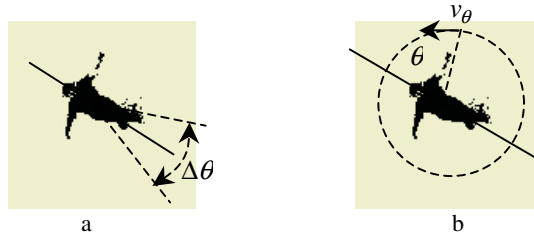
## 2 FOA Representation of Human Motion Activity

As it has already been mentioned, the objective is to characterize, for further recognition, human activities in different scenarios, with variable and realistic conditions that may occur, like low image contrast and resolution, different camera-target relative positions and viewpoints, occlusions of body parts during the movements, and the huge variability of people features and situations.

However, although the human activity recognition task in such conditions may seem unfeasible, it is well known that humans can guess what are the main or basic movements that a target is performing with a non very well defined image structure [2]. Thus, the underlying motion structure of the movement of a target can provide enough visual cues to allow the recognition of basic human body movements.

Keeping this fact in mind, a motion-based structure to characterize basic and general movements of the body is proposed, which has been built on twofold considerations: (a) use of optical flow, and (b) attention centered on the target.

Therefore, the idea is to describe the person movements with respect to some point of the body, assuming the person is being tracked and segmented out from the background. Thus, a previous tracking and target segmentation is performed, which provides us at each time information about the position of the target.



**Fig. 1.** (a) Expected variation of legs movement around the body centroid. (b) Mean optical flow in a given direction

The center of attention, or fovea center, will be situated at the centroid of the region corresponding to the segmented target. In order to refer the motion to the center of the focus of attention  $v_c(t)$ , the optical flow of the target pixels  $v_i(t)$  will be referred to the centroid of the target,

$$v'_i(t) = v_i(t) - v_c(t)$$

Therefore, the target motion with respect to the image coordinates will be compensated, and only the relative motion of the different parts of the target, with respect to the center of attention, will be represented. The objective is to have a qualitative description of the movement, without segmenting or identifying parts of the body, due to the fact that segmenting and tracking each part of the body is a complex and difficult process that cannot be solved in many situations.

Let us have a look to the figure 1. We can assume that the body parts are arranged around the body centroid, and that certain parts of the body usually move around a certain angular range  $\Delta\theta$  around the body centroid, for instance, the expected angular variation of the legs movements (figure 1a).

In order we can have a unique reference for all the angular directions with respect to the same origin, they can be referred to the vertical axis of a standing up person. An estimation of the vertical axis of the body can be obtained either by computing the principal axis of the target region, or calibrating the field of view of a static camera, determining the vertical direction with respect to the floor at every image point.

Let us represent the mean optical flow  $v_\theta(t)$ , at each time  $t$ , with respect to the centroid at a certain angular direction  $\theta$  (figure 1b), as:

$$v_\theta(t) = \frac{1}{N_\theta} \sum_{k \in P_\theta} v'_k(t)$$

with  $P_\theta$  being the set of target pixels  $(x_k, y_k)$  that are in the  $\theta$  direction with respect to the target centroid  $(x_c, y_c)$ , and  $N_\theta = |P_\theta|$  the number of target pixels in such direction.

Decomposing the flow  $v_\theta(t)$  in its normal and radial direction with respect to the target centroid, will provide an estimation of the relative motion of that part of the body with respect to the centroid in terms of radial (moving from or towards the centroid) and normal (moving in a perpendicular direction to the radius towards either up/left or down/right, depending on the area of the body where  $\theta$  is situated).

Representing all angular values of  $v_\theta(t)$  along time becomes a 2D signal  $foa(t, \theta) = v_\theta(t)$  named *focus of attention representation* (FOA) of the target flow. The FOA provides a description of the evolution of the mean flow of the target pixels at every direction  $\theta$  with respect to the target centroid. The FOA extracted from a temporal sequence of a tracked person will provide us information about the general movements of the different parts of the body, without having an exact knowledge about the position and motion of each part of the body.

Thus the FOA representation at a given time has the following properties:

- It is a focus of attention representation, inspired in foveal imaging, where the representation is built around a fovea point, in this case, the target centroid.
- It is an active technique based on focusing the attention on the tracked target.
- Provides an angular description of the target with respect to the fovea point.
- The information provided for each angle can be easily interpreted using the normal and radial components of the flow.

Thinking about the discrete form of the FOA representation, it can be further simplified by representing the mean flow of the target along a finite set of orientations;  $\theta_i$ ;  $i = 0, \dots, N-1$ , where the chosen orientations could integrate the mean flow of nearby directions, that is, at each time  $t$ , given an orientation  $\theta$ , the mean flow  $foa(t, \theta_i) = v_{\theta_i}(t)$ , can be expressed as an integration of a receptive field area around direction  $\theta_i$ . This receptive field area would cover a certain angular range around the direction. We can define the response of the receptive field area around  $\theta_i$  direction, as a Gaussian weighted mean of the FOA in the nearby directions, that is

$$foa(t, \theta_i) = \int foa(t, \theta) e^{-(\theta - \theta_i)^2 / 2\sigma_\theta^2} d\theta$$

where  $\sigma_\theta$  is the typical deviation of the Gaussian receptive field, determining the scope of the receptive field area around each direction. Receptive fields may overlap depending on the scope determined by the standard deviation.

Different types of body movements will activate different receptive fields in different ways, forming defined patterns characterizing basic movements like walking, rising/putting down arms, bending, sitting, etc. The response of the receptive fields forming the FOA representation at each time will provide us a way of identifying such a type of basic movements.

### 3 Segmenting the FOA Representation

The final aim of the FOA representation is to allow a recognition of human actions. Once we have a representation, in a given feature space, in order to facilitate the recognition tasks, a temporal segmentation of the body movements would be desirable, in order to decompose a certain human action in simple temporal units containing a unique type of basic body movements.

Other works, like [10], were also aimed at segmenting sequences of human activity to select key pose actions, in order to describe a higher level human activity descrip-

tion. The approach presented here is similar to this basic idea used in [10] about linear prediction, but using other two different concepts.

In order to segment the FOA representation along time, we will look for changes in the FOA representation along time in a similar way changes in video shot sequences are detected. The way changes are detected in the FOA are inspired in the work of [4] for video change detection, which uses the main motion present between two images of a sequence as a way to predict changes in the same video shot.

In a similar way, given the  $foa(t-1, \theta_i)$  values of a tracked target for the receptive fields  $\theta_i$  at a time  $t-1$ , we can predict the FOA response at a time  $t$ ,  $foa^*(t, \theta_i)$ . Thus, given the new measured  $foa(t, \theta_i)$ , we can define the following difference function  $Dfoa(t)$  to detect changes:

$$Dfoa(t) = \sum_i \left| foa(t, \theta_i) - foa^*(t, \theta_i) \right|$$

Looking for significant local maxima in the  $Dfoa(t)$  function, we can identify the times at which there is a noticeable change in the body movements performed by the target. Bear in mind that the values of  $foa(t, \theta) = v_\theta(t)$  are motion vectors of two components, expressed either in the Cartesian components or in the radial-normal components mentioned in the previous section.

To compute the estimate of  $foa^*(t, \theta_i)$  from  $foa(t-1, \theta_i)$ , the following approach is used. Given  $v'_k(t-1)$ , the vector field referred to the target centroid at time  $t-1$ , we can estimate the flow field at time  $t$  of every pixel belonging to the target at time  $t-1$ . Given the flow vector  $v'_k(t-1)$  of pixel  $p_k(t-1) = (x_k, y_k)$ , we can estimate the new position of pixel  $p_k$  in time  $t$  by

$$p_k^*(t) = (x_k^*, y_k^*) = p_k(t-1) + v'_k(t-1)$$

To the estimated position of the pixel  $p_k^*(t)$ , the flow vector  $v_k^*(t)$  estimated for time  $t$  at this position will be figured out by applying an uniform movement assumption, that is,  $v_k^*(t) = v'_k(t-1)$ . Therefore, the estimated mean flow field vector at time  $t$ , that is, the estimated FOA at time  $t$ , can be computed as

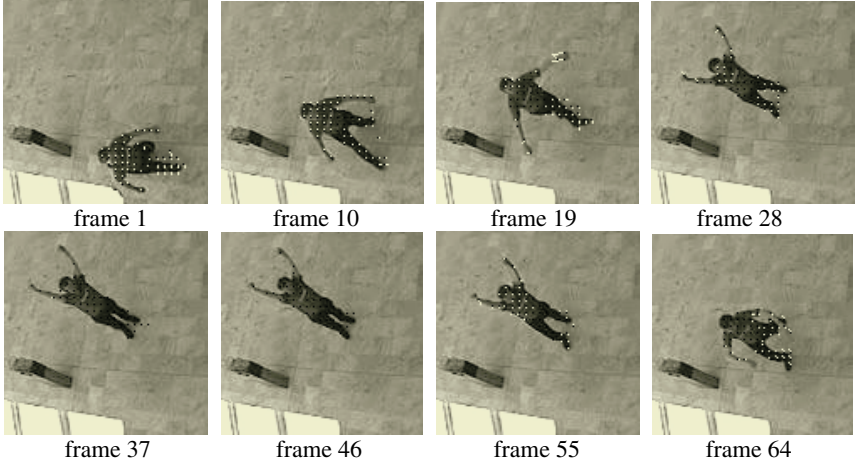
$$foa^*(t, \theta) = v_\theta^*(t) = \frac{1}{N_\theta} \sum_{k \in P_\theta} v_k^*(t)$$

with  $P_\theta$  being the set of target pixels  $(x_k, y_k)$  that are in the  $\theta$  direction with respect to the target centroid  $(x_c, y_c)$  at time  $t$ , and  $N_\theta = |P_\theta|$  the number of target pixels in such a direction.

## 4 Experiments and Examples

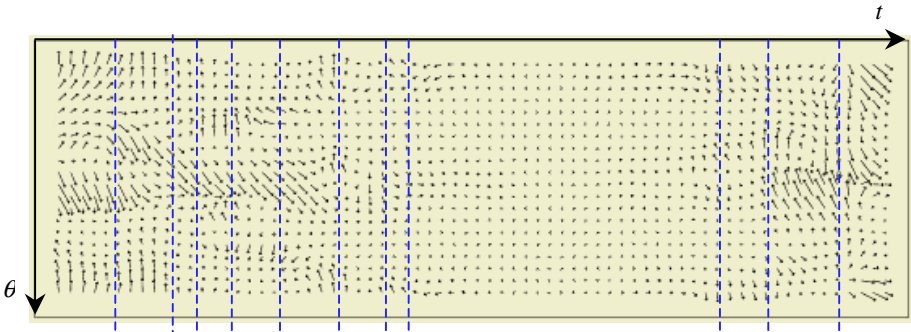
In order to see the effectiveness of the FOA representation and the performance of the FOA segmentation method introduced in section 3, the method has been tested using

some sequences of the CAVIAR project [11]. Figure 2 shows some frames of a sequence in a hall of a building entrance, where the tracked person performs a movement combining stepping, turning the upper part of the body and rising arms, afterwards, he stands by for a moment while the arms are up and then he comes back to the initial position.



**Fig. 2.** Some frames of a sequence of 70 frames of a target person.

Figure 3 shows the FOA representation of the 70 frames of the sequence in figure 2 using 20 receptive fields. In this case, the fields are placed every 18 degrees from the angle origin, which is placed at the head direction of the principal axis of the target. The principal axis at each frame  $t$  of the sequence has been estimated from the blob corresponding to the segmented target. The center of the FOA representation has been chosen as the centroid of the blob, which is also placed on the principal axis.



**Fig. 3.**  $foa(t, \theta)$  representation of the sequence in figure 6 using 20 receptive fields.

The flow vectors in figure 3 represent, at each time  $t$ , the mean flow computed by each receptive field  $\theta_i$ ;  $i = 0, \dots, N_\theta - 1$ , with  $N_\theta = 20$ . The flow vectors are expressed in terms of the normal and radial components with respect to the direction of the re-

ceptive field, that is,  $foa(t, \theta) = v_{\theta}(t) = (v_{R\theta}, v_{N\theta})(t)$ . The radial component  $v_{R\theta}$  of each vector is represented along the abscissas axis ( $t$  axis) and the normal component  $v_{N\theta}$  is represented along the ordinates axis ( $\theta$  axis).

Looking at figure 3, we can notice how the FOA presents differentiated patterns at different times, corresponding to the different movements of the parts of the body. For instance, the flow field in the first 5 frames corresponds to the activity present at the legs, that is, the receptive fields at the middle, which represents the stepping action of the person. We can even distinguish the movement performed by each leg in opposite direction; all measured with respect to the focus of attention center, that is, the centroid. We can also notice a movement in the upper part of the body, corresponding to the firsts and lasts receptive fields. This movement has a strong normal component that characterizes the turning movement of the upper part of the body the person is performing while stepping, in this case, the person is turning leftwards with respect to the principal axis of the body.

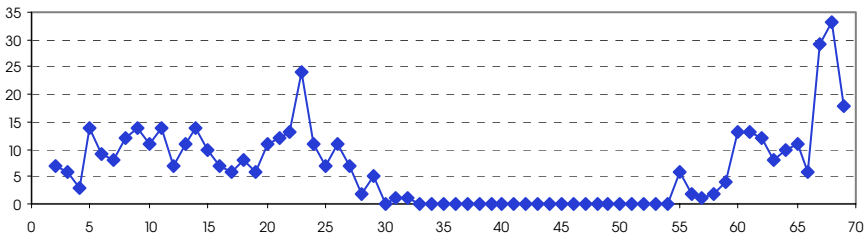


Fig. 4.  $Dfoa(t)$  of the FOA in figure 3.

Figure 4 shows the computed  $Dfoa$  of the FOA in figure 3, in order to segment the FOA representation in basic units with uniform motion values of the different parts of the target. The  $Dfoa$  has been computed using 72 receptive fields, that is, one every 5 degrees, and with a standard deviation of  $\sigma_{\theta}=1$  degree, that is, without no appreciable overlapping between receptive fields. The prediction was approximated by using the  $t-1$  segmented target instead of the segmented target at  $t$ , for the sake of computational efficiency. The local maxima of the  $Dfoa$  in figure 4 have been represented by dashed vertical lines in the corresponding representation of the FOA in figure 3 to illustrate how the segments between these limits show an uniformity in the motion values.

## 5 Conclusions and Further Work

This paper has described a technique to characterize human actions in visual surveillance scenarios in order to describe, in a qualitative way, basic human movements in general imaging conditions. The representation proposed is based on the introduced focus of attention approach, the FOA, building the representation from the point of view of the tracked target, thus becoming part of the active vision process to describe target movements. The introduced representation is based on motion information, particularly optical flow from respect to the fovea point.

The representation has been tested in some sequences from the database of the CAVIAR project, and the results obtained show its effectiveness to represent differentiate patters for different types of body moments, which could also be complex or combined movements of the different parts of the body.

The main further work is directed to apply some classification techniques to the FOA segments in order to identify and recognize automatically the sequence of basic movements.

## References

1. BenAbdelkader, C.; Cutler, R. and Davis, L.; "Motion-based recognition of people in EigenGait space"; V Int. Conf. on Automatic Face Gesture Recognition, 2002.
2. Bobick, A. F. and Davis, J.W.; "The recognition of human movement using temporal templates", IEEE. Trans. on PAMI, 23(3): 257-267, 2001.
3. Bodor, R.; Jackson, B. and Papanikoloupolos, N.; "Vision-based human tracking and activity recognition", XI Mediterranean Conf. on Control and Automation, 2003.
4. Bouthemey, P.; Gelgon, M. and Ganansia, F.; "A unified approach to shot change detection and camera motion characterization". IEEE Trans. on Circuits and Systems for Video Technology, 9(7):1030-1044, October 1999.
5. Bradski, G.R. and Davis, J.W.; "Motion segmentation and pose recognition with motion history gradients", Machine Vision and Applications, 13: 174-184, 2002.
6. Davis, J.W. and Gao, H.; "An expressive three-mode principal components model of human action style", Image and Vision Computing, 21: 1001-1016, 2003.
7. Davis, J.W. and Tyagi, A.; "A reliable-inference framework for recognition of human actions"; IEEE Conf. on Advance Video and Signal Based Surveillance, 169-176, 2003.
8. Essa, I.A. and Pentland, A.P.; "Coding, analysis, interpretation and recognition of facial expressions", IEEE Trans. on PAMI, 19(7): 757-763, 1997.
9. Masoud, O. and Papanikolopoulos, N.; "Recognizing human activities", IEEE Conf. on Advanced Video and Signal Surveillance, 2003.
10. Rui, Y. and Anandan, P.; "Segmenting visual actions based on spatio-temporal motion patterns", IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2000.
11. CAVIAR Project IST 2001 37540, <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>.

# Modelling Spatial Correlation and Image Statistics for Improved Tracking of Human Gestures

Rik Bellens, Sidharta Gautama, and Johan D’Haeyer

University of Ghent, Department of Telecommunications and Information Processing  
St.-Pietersnieuwstraat 41, B-9000 Gent, Belgium  
`rik.bellens@telin.ugent.be`

**Abstract.** In this paper, we examine sensor specific distributions of local image operators (edge and line detectors), which describe the appearance of people in video sequences. The distributions are used to describe a probabilistic articulated motion model to track the gestures of a person in terms of arms and body movement, which is solved using a particle filter. We focus on modeling the statistics of one sensor and examine the influence of image noise and scale, and the spatial accuracy that is obtainable. Additionally spatial correlation between pixels is modeled in the appearance model. We show that by neglecting the correlation high detection probabilities are quickly overestimated, which can often lead to false positives. Using the weighted geometric mean of pixel information leads to much improved results.

## 1 Introduction

Tracking humans is not an easy task. A system is needed that is general enough to capture all the variations in human appearance, but at the same time is specific enough to be able to distinguish between humans and other objects with similar structures.

Many methods have been proposed to track people. A survey can be found in [1] and in [5]. The complexity of the system depends on the desired level of detail in which the pose and movement of the human is described and of the a priori made assumptions about the appearance of people and background. For example, in surveillance applications, we might merely be interested in whether or not a person is present, in human-machine interfaces the machine should be able to read the gestures of the person and in virtual reality applications a complete three dimensional description of shape, pose and movement is needed to copy the gestures and movements as faithful as possible. Most of the present solutions constraint the environment by making some assumptions about the appearance of people and background. For example, they might assume a static background, people with special clothes, no moving objects other than humans, ... The system we will adapt is based on an article by Sidenbladh [7]. This system results in a three dimensional description of the pose of a person in every frame and is applicable in an unconstrained environment.



In this system the tracking is solved using the particle filter or condensation algorithm [2, 3]. Herein, an analysis-by-synthesis approach is followed. This means that first, a model of the human body, consisting of connected cones representing the individual limbs, is given a number of poses. Every pose is analysed through the Appearance Model and assigned a match measure. The best match for every frame is the desired output of the system. For the next frame new poses are synthesised based on the best poses from the previous frame and the expected change in pose modelled by the Temporal Model. For the first frame the correct pose of the human model is manually initialised.

The Appearance Model calculates a match measure between a state of the human model and the image information in a particular frame. In a formal way, the match metric is defined as the probability that a person in the given pose is present in the current frame and is calculated by comparing the actual image information, in our case edge and ridge responses, with the expected image information when a person in the given pose would be present. Originally, the image information is fused by neglecting correlations (1). To model the expected image information we learn the distributions of the filter responses, both on and off people. We use steered edge and ridge operators on different image scales ([7]).

$$p(F_i|\phi_t) = \kappa \prod_{x \in \text{foreground}} \prod_{i \in \text{cues}} \frac{p_{\text{on}}(f_i(x))}{p_{\text{off}}(f_i(x))}. \quad (1)$$

In this paper we present two improvements to this system: (i) sensor specific distributions and (ii) information fusing taking into account the correlation between the information. In [7] general distributions were established based on (high quality) training images found on the internet. By examining the statistics of one sensor, we are able to model sensor specific noise and blur, which leads to better performances. This is discussed in Sec. 2. In [7] information from different information sources, i.e. different cues and different spatial points, are fused in naive Bayesian fashion, assuming independence between these information sources. We show that this leads to very peaked distributions and will finally result in poor performance. By using the weighted geometric mean for fusing pixel information, where weights depend on the correlation between the pixel information, we are able to flatten these distributions, which leads to much improved results. This is discussed in Sec. 3. Section 4 will present some results of tracking experiments and in Sec. 5 our conclusions and some ideas for future work will be formulated.

## 2 Learning Sensor Specific Distributions

Distributions are learned by annotating a set of training images with the true location of limb boundaries. Edge and ridge responses are calculated on different scales, on and off edges and axes of limbs. Histograms are used to estimate foreground and background distributions of edge and ridge responses.

Obviously, the estimated distributions will depend on the set of training images used; the question is how much. The training images used by Sidenbladh

([7]) were collected from the internet, and were taken by different, high quality cameras. This training set, as well as the annotations of limb edges, can be found on the internet<sup>1</sup>. In this paper we investigate how suitable these distributions are when using cameras with other properties. We do this by comparing the general distributions with sensor specific distributions obtained with a set of roughly one hundred training images taken with a standard webcam. These images are typically noisy and blurry. Examples are shown in Fig. 1.



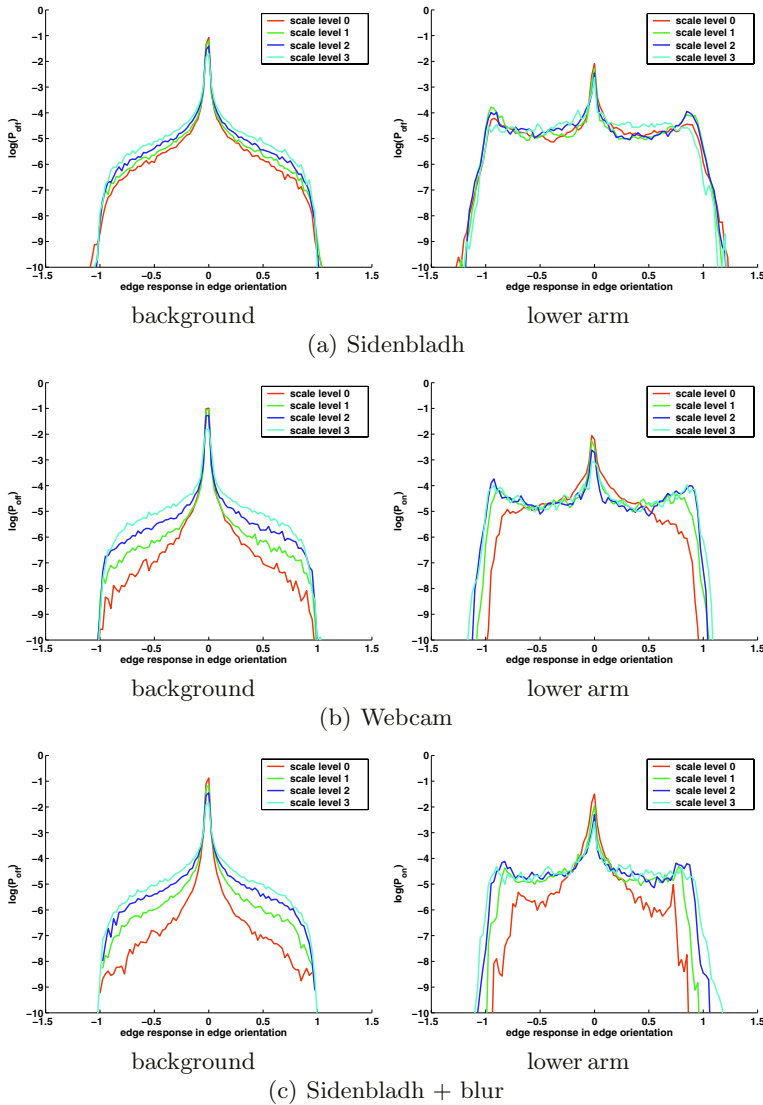
**Fig. 1.** Examples of training images for the webcam.

The top row of Fig. 2 shows the foreground and background edge distributions for the training set used by Sidenbladh, the middle row shows the distributions for the webcam training set. The distributions for the background show a maximum at 0, whereas those for the foreground show two extra submaxima at 1 and  $-1$ . As expected high edge responses (in absolute value) are more likely to belong to a point on the edge of a limb than to a point on the background. The difference between the distributions of the background and those of the foreground make it possible to distinguish between points on the foreground and points on the background. Similar results are obtained for ridge responses.

As can be seen, the distributions of the webcam are slimmer than those of Sidenbladh. Lower edge responses are observed. Additionally, where the distributions found by Sidenbladh are more or less equal on different image scales, the webcam distributions are clearly different on different image scales. In [6], Ruderman showed that the statistics of natural images are equal for different scale levels. We will show that this contradiction is mainly caused by the blurry images.

To examine the influence of blur on the distributions, we filter the training images of Sidenbladh repeatedly with a mean filter. The resulting distributions are shown in the bottom row of Fig. 2. These distributions are very similar with the distributions obtained with the webcam training set. For all scale levels, the peak around zero increases and the higher edge responses become less likely. For higher scale levels the influence decreases, this is due to the subsampling performed when generating the scale pyramid, which sharpens the images. Ruderman [6] observed that blur decreases the energy of high spatial frequencies and that the influence decreases for lower spatial frequencies, which is in accordance with our results.

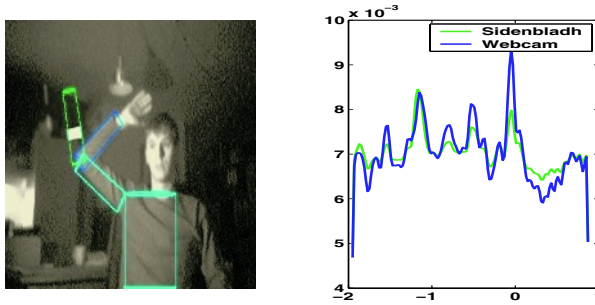
<sup>1</sup> <http://www.nada.kth.se/~hedvig/data.html>



**Fig. 2.** logarithm of the edge distributions of the background and the lower arm for different scales.

To examine the influence of the training set on the match metric we conduct the following experiment. The match measure between an image and a number of states of the human model are calculated, both with the Sidenbladh distributions and with the webcam distributions. These states all differ in only one parameter, the elbow angle. As can be seen in Fig. 3, the Sidenbladh distributions are easily misled by a small image structure which contains sharp edges. Even when the edges of a projected state coincide for only a small distance with the edges of

this structure, with the Sidenbladh distributions, this state will be assigned a high match metric because of the dominant character of high edge responses. With the webcam distributions, high edge responses are less dominant, which results in a higher match metric for a state of which the edges coincide over a longer distant with blurry image edges.



**Fig. 3.** The match measure calculated with the Sidenbladh distributions (green) and the webcam distributions (blue) for different poses with different elbow angles. The Sidenbladh distributions are misled by sharp edges, even when they only coincide over a small distance with the edges of the projected limb. The webcam distributions select the correct state.

### 3 Spatial Correlation

In [7] information is fused in a naive Bayesian way, assuming independence between different information sources. As we all know, natural images show high spatial correlations. Besides spatial correlation, we expect that there will also be correlation between the edge responses at different scales, since edges which are clearly visible on one scale, will also likely to be visible on other scales.

Suppose we are fusing information from  $N$  totally correlated information sources in a naive Bayesian way. In that case, the calculated measure is the  $N^{\text{th}}$  power of the true probability. As a result all probabilities are underestimated, but lower probabilities more than higher ones. Since match measures are normalized to one, this means that high probabilities will be overestimated. In real life cases the correlation will be less high, but high probabilities will still be overestimated. In the particle filter not only the best match is used to estimate possible states for the next frame, but all relative good matches are used depending on their match measure. This has the advantage that when the best match is not the correct state, due to shortcomings of the modelling of human appearance or due to occlusion, the correct state is not lost and the system can recover in the next frame. When one state is highly overestimated, all other states will be lost and the performance of the particle filter decreases a great deal.

To solve the problem of overestimation, we propose a new way to fuse the information from the different information sources by using the weighted geometric mean of marginal probabilities, rather than the simple product. We will

calculate the joint probability of  $N$  events  $A_i, i = 1..N$ , as the product of the marginal probabilities of these events with a correction exponent  $w$  to flatten the result:

$$p(A_1, A_2, \dots, A_N) = \prod_{i=1}^N p(A_i)^w \quad (2)$$

If  $w$  equals 1, we get the naive Bayesian approach, which is correct for total independency. If  $w$  equals  $\frac{1}{N}$ , we get the correct calculation in case of total dependency. The actual dependency will lie in between these two extreme cases. We will therefore use weights which lie in between those two extremes. The actual value of the weight  $w$  will depend on the average value of the correlation coefficient between the events  $A_i$   $\rho$  and the number of events  $N$  as follows:

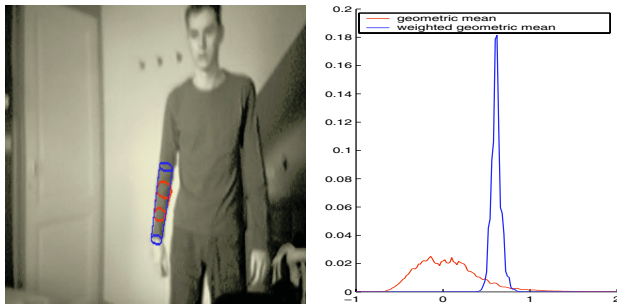
$$w(N, \rho) = \frac{1 + (N - 1)(1 - \rho)}{N} \quad (3)$$

In case  $\rho$  equals zero, we get the formulation for zero dependency, when  $\rho$  equals one, we get the formulation for total dependency. We can consider  $w$  as a metric of the amount of independent information present in the  $N$  events. The correlation coefficients are different for foreground and background, and different weights are used for different kind of information. First of all the information from neighboring points will be fused. We will chose  $N$  equally spaced points on the edge or ridge of a limb of a predicted state. For every point we will look up the marginal probability that it belongs to the foreground or the background in the learned distributions. The mean distance between the points is calculated and the average correlation coefficient is looked up in the learned correlation coefficient curve of foreground and background.

The average correlation coefficient depends on the mean distance between points. As a result, the amount of independent information will also depend on the length of the limb. A longer limb will result in higher weights. The human model we use is three dimensional. When a limb rotates perpendicular to the image plane, this can result in shorter projected limbs. The edges and ridges of these projected limbs might still coincide with the edges and ridges in the image, even when the correct state would have a longer projected limb. When using the naive Bayes approach and using a fixed number of points on each limb both states would be assigned the same match metric. When using our approach, a state which results in longer projected limbs, will have higher fusing weights and thus, when the edges and ridges coincide over the total distance, will have a higher match metric than states with shorter projected limbs. Our approach will thus not only increase the survival rate, but will also be able to distinguish better between good and bad states. This is shown in Fig. 4.

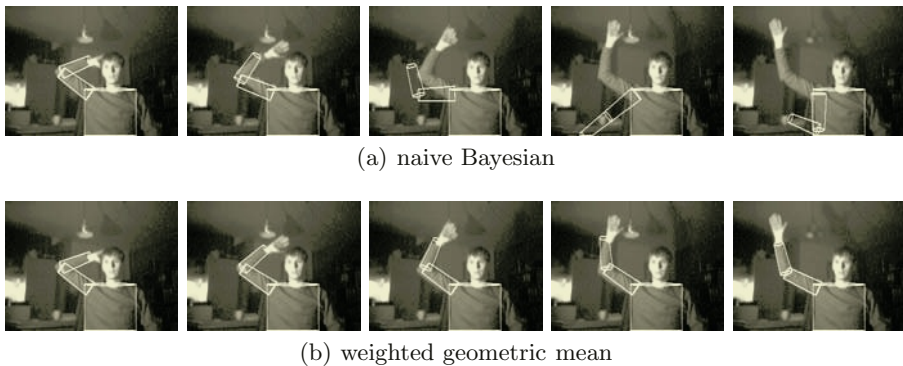
## 4 Tracking Experiments

In our first tracking experiment, we try to follow a waving arm using the edge and ridge cue. Only the arm of the person moves and only parallel to the image



**Fig. 4.** The match measure for different poses of the human model resulting in different projected arm lengths; in red the  $N^{\text{th}}$  root of the naive Bayesian measure, in blue the weighted geometric mean measure.

plane. As a result, we only need to estimate two parameters: one of the three shoulder angles and the elbow angle. All the other parameter have fixed values. This simplifies the experiment a lot. We tried tracking with naive Bayesian fusion and with the weighted geometric mean and used thereby 10 particles. The results are shown in Fig. 5. Although the estimation for certain frames is not very good, when using weighted geometric mean, we are able to recover from this. When using the naive bayesian method, once the estimation is no longer correct, the system can not recover. Using weighted geometric mean leads to more robust tracking.



**Fig. 5.** Frames 30, 40, 50, 60 and 70 of a tracking experiment of a waving arm (estimating 2 parameters) using 10 particles.

## 5 Conclusions and Future Work

In this work we have proposed two improvements to the original system. One, by using sensor specific distributions, the system can better distinguish the correct state. Two, by using dependency correction when fusing information the tracking

becomes more robust. Our system is able to track body parts in a fast and robust manner. When tracking a full body two problems occur: one, the correct state cannot always be distinguished and two, the computational time increases exponentially with the complexity of the human model. Solutions for the first problem can be found in using more image information, e.g. texture, (skin) colour, motion, . . . Two strategies can be followed for trying to solve the second problem. One, by using a more advanced temporal model, which is more specific to human motion, better prior predictions of a new state can be made and less energy is lost in exploring areas which will not lead to the correct state. Two, one might try to lower the number of needed particles by using a faster variations of the particle filter, like partitioned sampling [4].

## References

1. D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.
2. N. Gordon. A novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings on Radar, Sonar and Navigation*, 140(2):107–113, 1993.
3. M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
4. J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand-tracking. *European Conference on Computer Vision, ECCV*, 2:3–19, 2000.
5. T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.
6. D.L. Ruderman. Origins of scaling in natural images. *Vision Research*, 37(23):3385–3395, 1997.
7. H. Sidenbladh and M.J. Black. Learning the statistics of people in images and video. *International Journal of Computer Vision*, 54(1-3):183–209, 2003.

# Fast and Accurate Hand Pose Detection for Human-Robot Interaction\*

Luis Antón-Canalís<sup>1</sup>, Elena Sánchez-Nielsen<sup>2</sup>, and Modesto Castrillón-Santana<sup>1</sup>

<sup>1</sup> Institute of Intelligent Systems and Numerical Applications in Engineering  
Campus Universitario de Tafira, 35017 Gran Canaria, Spain

<sup>2</sup> Department of S.O.R. and Computation, 38271 University of La Laguna, Spain  
enielsen@ull.es

**Abstract.** Enabling natural human-robot interaction using computer vision based applications requires fast and accurate hand detection. However, previous works in this field assume different constraints, like a limitation in the number of detected gestures, because hands are highly complex objects difficult to locate. This paper presents an approach which integrates temporal coherence cues and hand detection based on wrists using a cascade classifier. With this approach, we introduce three main contributions: (1) a transparent initialization mechanism without user participation for segmenting hands independently of their gesture, (2) a larger number of detected gestures as well as a faster training phase than previous cascade classifier based methods and (3) near real-time performance for hand pose detection in video streams.

## 1 Introduction

Improving human-robot interaction has been an active research field in recent years in robotics community. A major challenge is based on detecting and interpreting human behaviours in video data, since it is essential for enabling natural human robot interaction. Our attention focuses on the communication with robots via hand gestures, which are a natural means of non-verbal communication for people.

In this paper, a fast and accurate hand pose detection approach that detects hand gestures in video streams for human-robot interaction is presented. In our approach, the hand pose detection problem is formulated in terms of the integration and combination of temporal coherence information and a cascade classifier method.

The cascade classifier method is based on the fastest and most accurate pattern detection approach for faces in monocular grey level-images [1]. This classifier is trained to detect wrists as an issue for locating hands in the first frames where the interaction with the machine takes place and as mechanism for system reinitialization. The main advantage of this approach is that wrists are highly independent from the gesture being made, so hands are detected without taking into account the gesture. Temporal coherence information is supplied by a template tracker with the aim of achieving real-time performance.

---

\* This work has been supported by the Spanish Government, the Canary Islands Autonomous Government and the Univ. of Las Palmas de G.C. under projects TIN2004-07087, PI20003/165 and UNI2003/06.



## 2 Related Work

Nowadays, there are several obstacles for achieving robust and efficient hand pose detection methods in video data, mainly due to the fact that the different posed difficulties such as variability and flexibility of articulated hand structure, shape of gestures, real-time performance, varying illumination conditions and complex background clutter. Therefore, previous works assume different constraints, like a limitation in the number of detected gestures using for example a watershed algorithm on the skin-like coloured pixel in collaboration with a particle filtering algorithm [2] for segmenting a specific set of hand gestures [3] or a no-real time hand detection against arbitrary background with an 86% accuracy rate through the use of an elastic graph matching technique for robot control [4]. Also, robust initialization and reinitialization must be addressed in order to carry out an effective hand pose estimation approach when a tracking method is used. However, most tracking approaches need to be manually initialized and cannot recover themselves when they lose the tracked target. As a result, some approaches often assume that the template which represents the target object is correctly aligned in the first frame [5]. Other approaches select the reference models by a hand-drawn prototype template, i.e., an ellipse outline for faces [6]. Moreover, the use of dynamic models that characterize hand motion such as particle filtering algorithm [2] requires training using the object moving over an uncluttered background to learn the motion model parameters before it can be applied to the real scene. However, transparent initializations without user participation are required for interactive human-robot communication.

Recent hand pose detection approaches are focused on Viola-Jones [1] cascade classifiers, commonly used for detecting faces. Although frontal faces share common features (eyes, eyebrows, nose, mouth, hair), hands are not so easily described. Their variability and flexibility make them highly deformable objects, so training a cascade classifier for detecting hands is a complex and arduous task. For that reason, a different classifier for each recognizable gesture has been trained [7], or a single classifier for a limited set of hands has been proposed [8]. However, the use of these approaches leads to the detection of a low number of gestures. Furthermore, real-time performance is not achieved with a cascade classifier method such as the one illustrated in [9] and only 15° rotations can be efficiently detected with a Viola-Jones detector [10]. Most importantly, the training data must contain rotated hand samples within these limits. Therefore, our approach changes the detection target to wrists. As a result, hands are detected without taking into account the gesture. Additionally, there is no limitation in the number of gestures being detected, as long as wrists are not occluded. Furthermore, fast computation is achieved incorporating temporal coherence information. And, the training time for the cascade classifier is greatly reduced. In the following sections, the proposed solution will be described and evaluated with experiments.

## 3 System Initialization

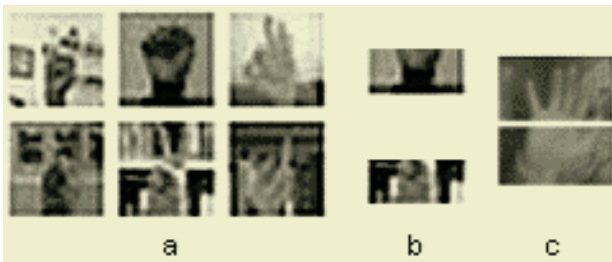
The Viola-Jones based cascade classifier [1] is used in order to automatically initialize the system for detecting hands in the first frames when the interaction takes place.

This cascade method combines increasingly more complex classifiers in order to quickly discard background areas on the first stages while a deeper analysis is performed in areas of high interest. Hands, however, are highly deformable objects, hard to train and classify due to their variability and flexibility. Next subsections describe the steps taken from the training of the classifier to the final hand extraction.

### 3.1 Training the Classifier

Training samples must be collected in order to train a cascade classifier. There are two categories: negative and positive samples. The first ones are related to non-object images while positive samples correspond to object images. However, the underlying problem with hand shapes in the training stage is that they are not self-containing objects, so patches of non-object images (background) are shown within positive samples. This makes the training stage harder and time consuming. Different hand samples are shown in figure 1.a. Due to the presence of background patches among positive samples, it is necessary a large collection of images showing hands in front of different sceneries. This, added to the variation of light conditions and hand postures in order to include every possible setting, results in a high computational cost of the training stage and an unreliable detection.

We propose a simplification of the classifier method, using wrist images as object samples. Wrists are much simpler objects, so the variability among samples is reduced and thus a faster training stage is achieved. Some used wrist samples are shown in figure 1.b, while figure 1.c illustrates the difference between the lower and upper section of a positive sample, being the former a simpler object. As long as wrists are not occluded, their detection leads to their hand, thus fulfilling the original goal.



**Fig. 1.** Positive sample images: a) whole hand, b) lower part of hands, used by our wrist classifier, c) detail of a sample image, divided in two sections.

### 3.2 Finding and Isolating Hand Pattern

In order to reduce the search space where the wrist cascade classifier is applied, people is first located using a cascade classifier as described in [11]. According to average human body proportions [12], an arm length is around three times the length of a head, so a boundary of the distance that a hand can reach knowing the location of a head can be computed. The result is that, for typical desktop images, more than a half of the original image may be removed from the problem space. If no faces are detected, the search space problem is aimed to the original image dimension.

Once a wrist has been located, its enclosing area, given by the cascade classifier, is resized in order to include the whole hand taking into account natural hand-wrist proportions, where a hand's height is between 2.9 and 3.1 times its wrist's width. This whole hand area is supplied as the tracking pattern, and will be followed in successive frames.

## 4 Hand Gesture Detection System

Our hand gesture detection system is based on a continuous operation that combines the results computed by the cascade classifier method with a template tracking module. The tracking module is used in order to get benefits from temporal coherence of the hand detection information provided by previous frames.

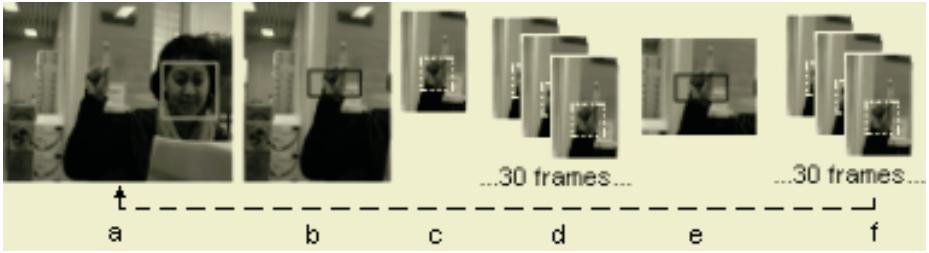
Robustness to background clutter and low computational costs are the main issues that need to be addressed when a tracker module is used in order to follow hands from previous frames. With this aim, we make use of the tracking algorithm of [13] that has been previously applied to different visual applications such as face and vehicle tracking. This algorithm is focused on the framework of representation spaces based on second order isomorphism [14] that allows the definition of *context objects* notion. The use of this concept allows taking into account similar objects related to the target object and deciding when it is necessary to update the target pattern. Updating hand patterns using this concept avoids confusing the tracked target with clutter and similar objects from background.

Our cyclic operating approach involves four different processing stages. The first process begins when the classifier finds a wrist in the way described in section 3.2. The hand it belongs to is selected as the tracking pattern, so in a second stage the tracker will follow it during the next 30 frames or until the pattern is considered lost. On the next stage, the wrist cascade classifier is applied again. This second time, however, the search space is reduced to an area close to the last tracked pattern position. Once again, the result of the classifier is selected as the new tracking pattern, which will be followed again during 30 more frames or until the pattern is considered lost, as it was achieved in the second stage. Finally, the operation cycle is restarted with a new application of the wrist cascade classifier on the whole search space, as described in section 3.2.

The main assumption underlying this approach is that hands can be frequently expected to enter and exit under view and that a robust reinitialization is required when the tracked hand is lost due to exceptional circumstances based on drastic appearance transformations in the gesture being made. An overview of the different processing stages that take place in our framework is shown in figure 2.

## 5 Experimental Results

In order to carry out empirical evaluations of the system, 12 different video streams with an average of 1500 frames each one, 320x240 pixels each frame, were acquired at 25 frames per second, and analyzed using a PIV 2.8 GHz. These videos contain 12 different people with assorted background and light conditions, making more than 20 vertical hand gestures. The first two subsections describe the results computed with the classifier method, analyzing the training stage and the performance of the classi-



**Fig. 2.** Hand Gesture Detection System: a) faces are detected, b) wrists detection in the reduced search area, due to faces detected, c) hand used as the tracking pattern, d) hand tracked during 30 frames, or pattern lost, e) new wrist detection, in space around last tracked position f) new detection tracked during 30 more frames, or pattern lost. Finally, the continuous cycle restarts in stage *a*.

fier method using wrists and using the whole hand. The last subsection shows the performance achieved when a tracker module is incorporated.

## 5.1 Training Stage

The used training set consists of 5653 negative samples and 4130 20x20 positive samples from our own dataset and other samples selected from available datasets [15]. The trainer only takes into account the lower part of those images, 20x10 pixels, which show a hand from its wrist to half the palm, including fingertips of flexed fingers and thumbs (both flexed and stretched), as shown in figure 1.b.

The first advantage of our wrist detector over a whole hand detector is the time needed for training. Using the same amount of training images, it takes less than 24 hours on a PIV 2.8 GHz to train an 18 stages classifier, while the hand classifier needs more than a week to train the same number of stages. Mainly because the variability of the lower half of a hand is much lower than that of a whole hand, so the classifier is able to find similarities among samples much faster.

## 5.2 Classifier Performance

Besides the lower training time, the wrist detector also reduces three times the false detection rate given by the hand detector. The high amount of gestures, background, people and light conditions present in sample images lead to an unreliable classifier. Using the same positive sample images, but taking into account only the lower part of them, simplifies the problem and therefore reports a false detection rate reduction.

The wrist detector, without the aid of the tracking module, was applied on the test video set. An average detection rate, in relation to the total number of frames, of 0.88 was achieved. This rate is not higher because the training set was originally created having a hand detector in mind, so it is not optimal for the training of wrists.

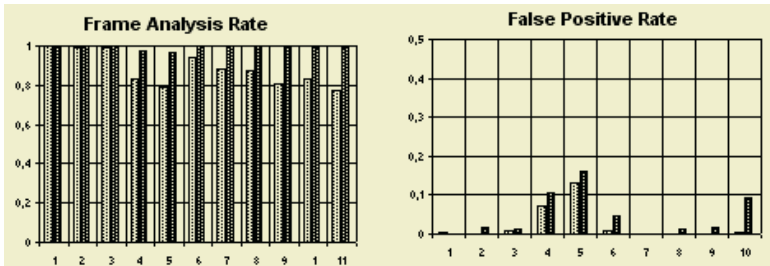
From the amount of frames where wrists were detected, we measured a 0.97 correct detection. Figure 3 illustrates different results using the wrist detector approach for isolating hand patterns with diverse people, background and light conditions.



**Fig. 3.** Hand pose detection results showing wrists detections (dark rectangle) and complete hands (white rectangle).

### 5.3 Tracking Influence

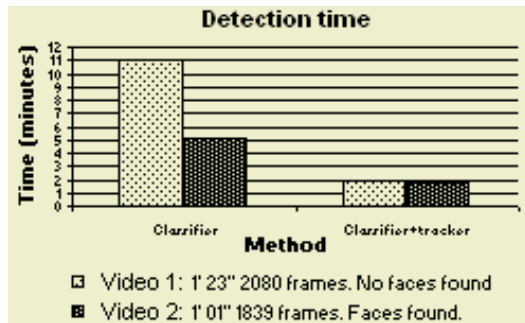
A second set of tests were performed combining the wrist detector and the pattern tracker, as described in section 4. Even though the false detection rate raises from 0.03 to 0.06, the amount of frames with detection also raises, from 0.88 to 0.99. There is an absolute increase of true positives of 7%. Figure 4 shows individual results in each video using both techniques. Figure 5 illustrates some frames from a video stream where the tracker follows a hand, which is both moving and changing gestures. The pattern size used for the tracking process is established to 24x24 pixels.



**Fig. 4.** Frame Analysis and False Positive Rate. Results of the classifier used alone in lighter bars, while darker bars show results for the classifier and tracker cycle.



**Fig. 5.** Four frames (210, 220, 230, and 240) from a video sequence, where the centre of the tracked pattern is represented by a cross. The rectangle corresponds to the whole hand area computed through the use of hand-wrist proportions from the last wrist detected using the classifier.



**Fig. 6.** Time measured for computing the classifier method and the time measured for computing the combination of classifier method and the tracker module for two different videos. Face location is not significant in relation to speed performance, when combining the classifier and the tracker module.

The average processing rate using the hand gesture detection system proposed in section 4 is 16 fps, while the average processing rate using only the classifier method reaches a maximum of 5 fps (2 fps when faces are not found). Figure 6 illustrates the measured time for two different videos using only the classifier method and using the classifier method plus the tracker module. Integrating a tracking module with a classifier based on wrists increases the speed achieved in previous works in relation to real-time performance [7, 8, 9] and also the number of different gestures detected. From these results, we have observed that the influence of face detection and the consequent search space reduction is significant when the classifier is used without the aid of the tracker. If the classifier method and the tracker module are combined, detecting faces is only significant in order to reduce false positives.

## 6 Conclusions and Future Work

We have developed a fast and robust hand pose detector that integrates temporal coherence information and a wrist detector using a continuously operational system.

We have tested our approach in different experiments which cover diverse people, backgrounds and light conditions. Two major conclusions have been obtained from the experiments: (i) the classifier method based on wrists reduces the false detection rate and the training stage in comparison to a whole hand detector and (ii) combining temporal coherence information and a classifier method based on wrist reduces greatly the hand pose detection time in respect to previous works based on classifier methods [7, 8, 9]. Moreover, the number of gestures detected is also increased.

Future research is focused on an improvement of the training set, estimating transition states of the hand gestures over the time with the purpose of only interpreting a new gesture, when it has taken place.

## References

1. Paul Viola and Michael J. Jones.: Rapid object detection using a boosted cascade of simple features. *IEEE Computer Vision and Pattern Recognition*, Volume 1, pp. 511-518, December 2001.

2. M. Isard and A. Blake: Condensation - Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5-28, 1998.
3. L. Brethes, P. Menezes, L.Lerasle and J. Hayet: Face tracking hand gesture recognition for human-robot interaction. *IEEE International Conference on Robotics and Automation*. New Orleans, April 26 – May 1, 2004.
4. J. Triesch and C. von der Malsburg: A System for Person-Independent Hand Posture Recognition against complex backgrounds. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(12):1449-1453, December 2001.
5. J.M. Rehg and T. Kanade: Visual tracking of high DOF articulated structures: an application to human hand tracking. In 3<sup>rd</sup> Proc. European Conference on Computer Vision, Volume II, 35-46.
6. M. Spengler, B. Schiele: Towards robust multi-cue integration for visual tracking. *Machine Vision and Applications*, Springer-Verlag 14: 50-58, 2003.
7. B. Stenger, A. Thayananthan, P. Torr and R. Cipolla: Hand Pose Estimation using Hierarchical Detection. In *ECCV Workshop on HCI 2004*, Lecture Notes in Computer Science, Springer-Verlag, vol. 3058, pp. 102-112.
8. Mathias Kösch and Matthew Turk: Robust hand detection. In 6<sup>th</sup> IEEE International Conference on Automatic Face and Gesture Recognition. May 17-19, 2004, Korea.
9. J.Barreto, P. Menezes and J. Dias: Human-Robot Interaction based on Haar-like Features and Eigenfaces. *IEEE International Conference on Robotics and Automation*. New Orleans, April 26 – May 1, 2004.
10. Mathias Kösch and Matthew Turk: Analysis of Rotational Robustness of Hand Detection with a Viola-Jones Detector. In *IAPR International Conference of Pattern Recognition*, 2004.
11. H. Kruppa, M. Castrillón and B. Schiele: Fast and Robust Face Finding via Local Context. In *Joint IEEE International Workshop on VS\_PETS*, Nice, France 2003.
12. S. Rogers Peck: *Atlas of Human Anatomy for the Artist*. Oxford University Press, Inc, USA, 1982. ISBN: 01950309858.
13. C. Guerra Artal: Contributions to visual precategory tracking. Phd thesis, University of Las Palmas G.C, 2002.
14. S. Edelman.: *Representation and Recognition in Vision*, The MIT Press, 1999.
15. J. Triesch. Hand Posture Database I, II. <http://www.idiap.ch/~marcel/Databases>

**Part VII**

**Surveillance**



# Performance Analysis of Homomorphic Systems for Image Change Detection\*

Gonzalo Pajares<sup>1</sup>, José Jaime Ruz<sup>2</sup>, and Jesús Manuel de la Cruz<sup>2</sup>

<sup>1</sup> Departamento de Sistemas Informáticos y Programación, Facultad de Informática  
Universidad Complutense de Madrid, 28040 Madrid, Spain  
pajares@dacya.ucm.es

<sup>2</sup> Departamento de Arquitectura de Computadores y Automática, Facultad de Informática  
Universidad Complutense de Madrid, 28040 Madrid, Spain  
{jjruz, jmcruz}@dacya.ucm.es

**Abstract.** Under illumination variations image change detection becomes a difficult task. Some existing image change detection methods try to compensate this effect. It is assumed that an image can be expressed in terms of its illumination and reflectance components. Detection of changes in the reflectance component is directly related to scene changes. In general, scene illumination varies slowly over space, whereas the reflectance component contains mainly spatially high frequency details. The intention is to apply the image change detection algorithm to the reflectance component only. The aim of this work is to analyze the performance of different homomorphic pre-filtering schemes for extracting the reflectance component so that the image change detection algorithm is applied only to this component. This scheme is not suitable for scenes without spatial high frequency details.

## 1 Introduction

The main difficult in image change detection tasks is the illumination variations between two frames. Some existing image change detection methods try to compensate this effect by mapping data and contextual information under an energy function which is then minimized through optimization [1,2,3].

Additionally some works have used the power of homomorphic systems in order to separate the reflectance component, so that the image change detection algorithm is only applied to this component [4]. This is justified under the assumption that scene illumination varies slowly over space, whereas the reflectance component contains mainly spatially high frequency details.

The goal of this work is to analyse the behaviour of three different homomorphic filtering strategies in image change detection. They are: the low pass filtering strategy given in [4] (TOT), the frequency procedure based on butterworth filtering [5,6] (KOV) and the wavelet-based approach described in [7] (GOM).

---

\* This work has been partially supported by the Spanish CICYT under grant DPI2002-02924.

We have selected the image change detection algorithm described in [1] as the base method. It is applied with and without previous homomorphic filtering, where the homomorphic filtering is implemented according to TOT, KOV and GOM.

This paper is organised as follows: in section 2 we formulate the homomorphic framework and describe the TOT, KOV and GOM strategies. In section 3 the homomorphic performance is analyzed. Finally in section 4 the conclusions are presented.

## 2 The Homomorphic Framework

The goal for image change detection between two frames is to obtain equal illuminated frames by processing them in any way. This can be achieved through a special class of systems know as *homomorphic systems* [5]. They are based on the image perception. The images people perceive consist of light reflected from the objects. The basic nature of intensity may be characterized by two components: (1) the amount of source light incident on the scene being viewed and (2) the amount of light reflected by the objects in the scene. They are called the *illumination* and *reflectance* components and are denoted by  $i_k(x,y)$  and  $r_k(x,y)$  respectively, where  $k$ -th frame and  $(x,y)$  is the pixel location. As a first approximation for Lambertian objects surfaces the intensity of the  $k$ -th frame in an image sequence is given by

$$f_k(x,y) = i_k(x,y)r_k(x,y) . \quad (1)$$

The illumination component of an image is generally characterized by slow spatial variations, while the reflectance component tends to vary abruptly, particularly at the junctions of dissimilar objects. These characteristics lead to associating the low frequencies with illumination and the high with reflectance.

The goal is to extract the reflectance component in order to minimize the illumination effects and to consider only the reflectance. This is carried out by first applying the logarithm and then extracting the high frequencies. The logarithm transforms the multiplicative relation in (1) into an additive one:

$$y_k = \log(f_k(x,y)) = \log(i_k(x,y)) + \log(r_k(x,y)) . \quad (2)$$

Although the log-nonlinearity modifies the spectral content of illumination and reflectance components, it is in practice often justified to assume the log-illumination to be still spatially slowly varying [5]. Obviously, this scheme should not be suitable for scenes without spatial high frequency details.

### 2.1 KOV: Through High-Pass Filtering

In [5,6] the homomorphic system is designed as follows: after applying the logarithm to  $f_k$  in (1), the resulting image  $y_k$  is fast Fourier transformed, the resulting image is high-pass filtered in the frequency domain by designing a high-pass filter based on the butterworth scheme. The filter function is obtained so that it affects the low- and high-frequency components of the Fourier transform. A trade-off must be achieved to

boost the high frequencies relative to the low frequency values. We have chosen this relation as a ratio of 25% and 75% for low and high frequencies respectively. So the filter function tends to decrease the low frequencies and amplify the high frequencies. Once the filtering is carried out, the inverse fast Fourier transformation is applied to the filtered result. Exponentiation of the last resulting signal is applied to obtain the reflectance image.

## 2.2 TOT: Through Low-Pass Filtering

In TOT [4] the homomorphic system works as follows: after applying the logarithm to  $f_k$  in (1), the resulting  $y_k$  of (2) is low-pass filtered and then subtracted from the logarithmic original  $y_k$ , yielding a high-pass component. The low-pass filtering can be carried out through different filter kernels; we have used a Gaussian one with size  $31 \times 31$  and standard deviation of 14 as in [4].

This filter has to trade-off the reduction of illumination and the retention of the image structure. Note that the standard deviation is decisive in the filter design. A high value of this parameter involves a high filter dimension affecting the high components in the image.

Exponentiation of the filtered signal is applied to obtain the reflectance image. In the reflectance image illumination effects are strongly suppressed while object information is preserved. In the illumination image, however, the light-spot is very prominent whereas object details are blurred. Of course, the illumination image still contains low-frequency parts from the reflectance and thus separation of the two components is only an approximation. Nevertheless this suffices for the intended image change detection.

## 2.3 GOM: Wavelet-Based Filtering

This is the scheme described in [7]. As before, the goal is to obtain the reflectance component. The process is as follows: first of all we must separate the illumination and reflection components. The logarithm applied to  $f_k$  converts the product from (1) to  $y_k$  which is a sum with two components that are low pass and high pass respectively. Then, these components are separated by using the Discrete Wavelet Transform (DWT). The DWT performs a low and high pass filtering using the scheme proposed by Mallat [8]. The scheme can be repeated over the approximation image. So we get a multiresolution approximation to the original image. The more we repeat the decomposition scheme the more we concentrate the low frequencies energy in the approximation image. After several decompositions (depends on image size) the approximation image is a representation of illumination of the image. In our approach we have used a five level decomposition.

Now, the illumination is in the approximation image of the DWT. If we want to obtain the same illumination in the both frames under processing, the next step is to cancel the approximation image and recover the whole image without illumination, i.e. only with the reflection component.

So, after the five level decomposition we replace the approximation image with a zero image of the same size that the approximation image at such level. After this replacement we apply the Inverse Discrete Wavelet Transform (IDWT) scheme using the reconstruction filters.

The resulting image is now with no illumination. Now a constant illumination level is added. This constant level is set  $\log(128)$  that corresponds to the mean grey level in the byte images representation used in this work. Then, the image can be recovered, i.e. the logarithm is undone by using the exponential. As before the exponentiation could be avoided for the image change detection purposes.

### 3 Comparative Analysis and Performance Evaluation

In order to analyse the performance of the *homomorphic systems*, we have considered two different data sets: a synthetic data set artificially generated and a real data set corresponding to consecutive frames in indoor and outdoor environments where the images are captured under different illumination conditions.

#### 3.1 Synthetic Data Set

We used as the image reference a region of  $400 \times 400$  pixels from a remote sensing image acquired by the commercial IKONOS satellite. This image contains mainly high frequency components and was assumed to be the  $t_1$  reference image in the full sequence (i.e. the image acquired at time  $t_1$ ).

Then we generate five new synthetic images  $t_i$ ;  $i = 2, \dots, 6$  from  $t_1$  by adding changes, noise and illumination variations as follows:

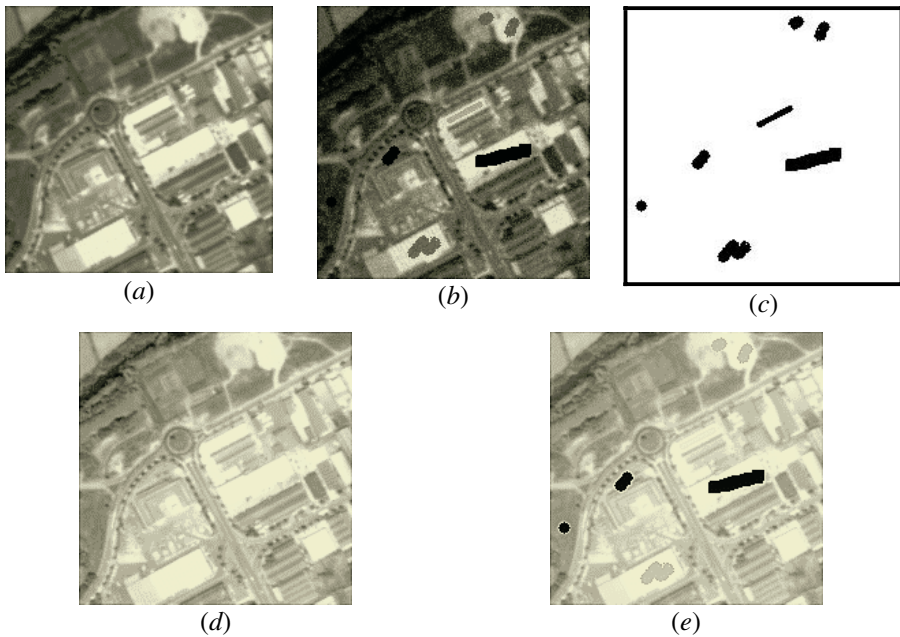
- $t_2$ : only controlled changes without noise and without illumination variation;
- $t_3$ : Gaussian noise ( $\sigma^2=2.5$ );
- $t_4$ : Gaussian noise ( $\sigma^2=5$ ) and illumination variation;
- $t_5$ : salt and pepper noise (density = 0.05);
- $t_6$ : salt and pepper noise (density = 0.10) and with illumination variation.

Hence, five pairs of images are built with each  $t_i$  and the reference  $t_1$ .

The illumination variation is achieved by shifting the original histogram, so that different light conditions can be simulated between the image with the original histogram and the image with the shifted histogram. Figure 1(a) and (b) show the  $t_1$  and  $t_4$  images. Figure 1(c) shows the mapping of the controlled changes. Figures 1(d) and (e) are obtained from (a) and (b) respectively by applying the homomorphic KOV scheme. We can see that the illumination is compensated between both images, i.e. the differences in the original images are minimized.

#### 3.2 Real Data Set

Real data sets for aerial or satellite images are only available in dedicated companies which have previously paid the corresponding royalty.



**Fig. 1.** Synthetic data set used in the experiments. (a)  $t_1$  original image, (b)  $t_4$  image with controlled changes and Gaussian; (c) map of the areas with simulated changes used as the reference map in the experiments; (d) and (e) illumination compensation by homomorphic filtering

Hence, the real data set used in the experiments consisted of a first group of ten pairs of indoor images captured under different illumination conditions. Indeed, we have varied the illumination in two ways:

*a*) by changing the internal artificial illumination switching on and off different indirect lights,

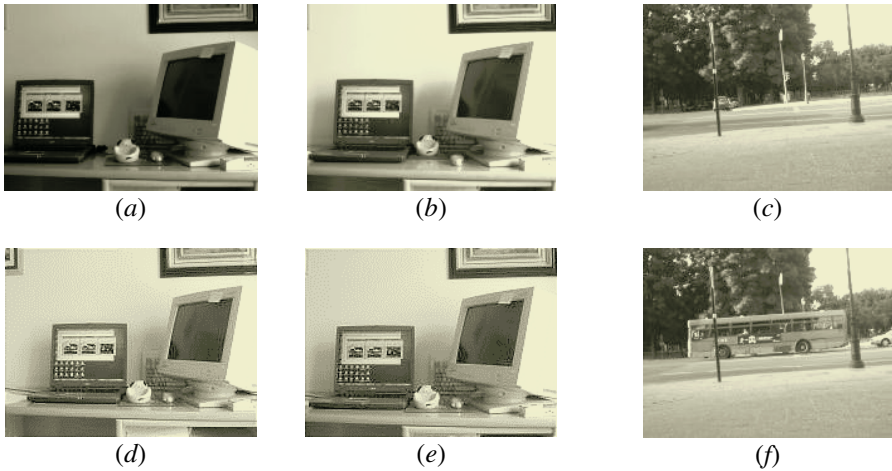
*b*) by moving a blind window, i.e. by changing the external illumination conditions.

Two representative images of this kind of data are shown in the figure 2(a) and (b). In the image (b) we have changed the mobile computer position and the two little objects placed between the mobile computer and the big monitor. Moreover, the image (b) is captured by increasing both, the internal and external illumination conditions. Figures 2(d) and (e) are obtained from (a) and (b) respectively by applying the homomorphic KOV approach. We can see once again that the illumination is compensated between both images.

The second group of real data is captured from an outdoor environment and consists of eight pairs of images; Figures 2(c) and (f) show two representative images. In this group the illumination is similar as they are captured in the same instant of time.

### 3.3 Evaluation

To evaluate the performance quantitatively, we used the change detection approach described in [1] and define the correct detection rate (CDR) and the false alarm rate (FAR) as follows [9].



**Fig. 2.** Real data sets: (a) and (b) original indoor images under different illumination conditions; (d) and (e) illumination compensation by homomorphic filtering; (c) and (f) original outdoor images under similar illumination conditions

CDR: the probability of claiming an Area of Interest (AOI) is changed when AOI is actually changed or claiming AOI is unchanged when AOI is actually unchanged.

FAR: the probability of claiming AOI is changed when AOI is actually unchanged or claiming AOI is unchanged when AOI is actually changed.

We have started our experiments with the synthetic data sets, because the changes are well controlled. Figure 3(a) and (b) shows graphically the percentage of CDR and FAR results for each pair of images obtained through the change detection method in [1] with homomorphic filtering (KOV, TOT and GOM) and without homomorphic filtering (WHF).

From results in figure 3, we can infer the following conclusions:

1. The best performance is achieved when KOV is applied, particularly for images including differences in the illumination (pairs 3 and 5).
2. In noisy images the homomorphic filtering does not contribute to the improvement of the results.
3. The worst results are obtained by GOM; we have verified that this is due to the removing of the approximation coefficients in the last decomposition level. This implies that in the reconstruction process appears some kind of artifacts, affecting the changes.

Now, taking into account the best performance achieved by using the KOV homomorphic filtering scheme, we have processed the real data set (indoor and outdoor) considering the results obtained under this filtering as the reference results for comparison purposes with the remainder homomorphic filtering schemes. We have averaged the CDR and FAR percentages over each set of data groups. Table 1 summarizes the results obtained for the indoor and outdoor environments for each homomorphic filtering scheme.

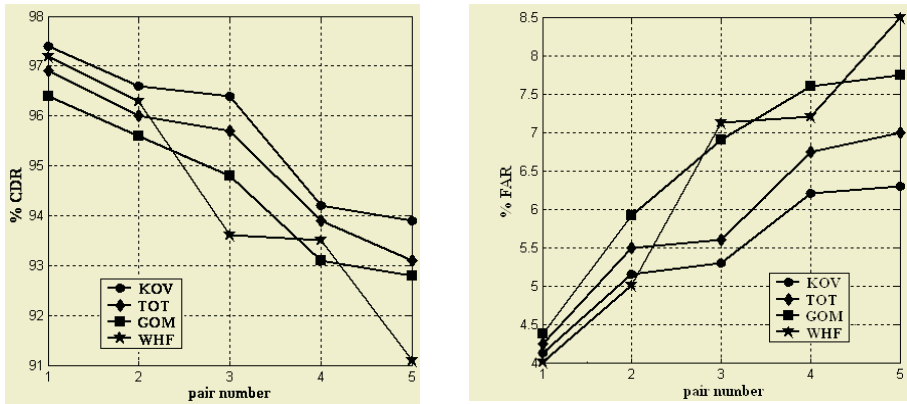


Fig. 3. Behaviours of the change detection error (%) with (KOV, TOT, GOM) and without (WHF) homomorphic filtering; (a) and (b) CDR and FAR respectively against the pair number

Table 1. Averaged CDR and FAR percentages for indoor and outdoor image sequences

	indoor environment		outdoor environment	
	CDR	FAR	CDR	FAR
KOV (reference)	100	0	100	0
TOT	92.86	6.67	99.21	1.12
GOM	88.12	9.87	96.66	2.46
WHF	83.32	14.01	98.51	1.11

From results in table 1, the following conclusions can be inferred:

1. In the indoor images, including changes in the illumination, the homomorphic filtering improves the final results.
2. There are still artifacts affecting the performance when GOM is used.
3. In the outdoor images, without significant illumination changes, the homomorphic filtering is irrelevant. Only some slight improvement is achieved, but without filtering the results are equally acceptable in the outdoor environment.

## 4 Conclusions

We have shown the performance of the homomorphic filtering for image change detection in image sequences. This is particularly valid with images displaying high illumination variability, i.e. for scenes with spatial high frequency details. We have also verified that the underlying noise in the images does not affect the final results. This works provides the guidelines for applying homomorphic filtering for change detection methods.

## References

1. Aach, T., Kaup, A.: Bayesian algorithms for adaptive change detection in image sequences using Markov Random fields. *Signal Processing: Image Communication*. 7 (1995) 147-160

2. Bruzzone, L., Fernández-Prieto, D.: Automatic Analysis of the difference Image for unsupervised change detection. *IEEE Trans. Geoscience Remote Sensing*. 38(3) (2000) 1171-1182.
3. Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: A Systematic Survey. Submitted to *IEEE Trans. Image Processing*, (available on-line) <http://www.ecse.rpi.edu/homepages/rjradke/pages/research.html>, 2004
4. Toth, D., Aach, T., Metzler, V.: Bayesian Spatio-Temporal Motion detection under varying illumination. In: M. Gabbouj, P. Kuosmanen, (eds.): *Proc. European Signal Processing Conference (EUSIPCO)*, Tampere, Finland, (2000) 2081-2084
5. Gonzalez, R.C. and Woods, E.R. 1993, *Digital Image Processing*. Addison-Wesley, Reading, MA (1993)
6. Kovesi, P.: MATLAB functions for Computer Vision and Image Analysis (available on-line) <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns.tar.gz> (2004)
7. Gómez-Moreno, H., Maldonado-Bascón, S., López-Ferreras, F. Martín.Martín, P. and Villafranca-Continente, J.M. 2000. Motion detection using support vector machines. In: *Proc. International Conf. Signal Processing and Communications* (available on-line) [http://www2.uah.es/teose/webpersonal/Hilario/Personal/Pagina\\_files/Publications.html](http://www2.uah.es/teose/webpersonal/Hilario/Personal/Pagina_files/Publications.html)
8. Mallat, S.: A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 11(7) (1989) 674-693
9. Liu, S.C., Fu, C.W., Chang, S.: Statistical Change Detection with Moments under Time-Varying Illumination. *IEEE Trans. Image Processing*. 7(9) (1998) 1258-1268



# Car License Plates Extraction and Recognition Based on Connected Components Analysis and HMM Decoding\*

David Llorens, Andrés Marzal, Vicente Palazón, and Juan M. Vilar

Dept. de Llenguatges i Sistemes Informàtics  
Universitat Jaume I, Castelló, Spain  
{dllorens, amarzal, palazon, jvilar}@lsi.uji.es

**Abstract.** A system for finding and recognizing car license plates is presented. The finding of the plates is based on the analysis of connected components of four different binarizations of the image. No assumptions are made about illumination and camera angle, and only mild assumptions regarding the size of the plate in the image are made. Recognition is performed by means of Hidden Markov Models. Experiments on a database of Spanish number plates show the feasibility of the proposed approach.

## 1 Introduction

Car License Plate Recognition (CLPR) has a wide variety of applications [1, 2], such as control of parking lots, borders or traffic, recovery of stolen cars, etc. Many of the current CLPR systems work under controlled light settings and assume that the plate is horizontal and/or perpendicular to the camera direction. We present a CLPR system that makes no assumptions about illumination and it makes only mild assumptions about the position of the camera and the relative size of the plate in the image. Figure 1 shows some typical images from our database.

Two different problems are faced when building a CLPR system: (1) License plate extraction: finding the area of the image that corresponds to the plate; and (2) recognizing the characters in the plate.

In our approach, the license plate extraction phase produces an ordered series of regions of interest (ROI). These regions are found by analyzing the connected components of four different binarizations of the image. Character recognition is performed on each ROI by means of a Hidden Markov Model (HMM) decoding system. The recognition yields a string of characters and an estimation of the probability, according to the HMMs, that those are the characters present in the text in the ROI. This value is used to rescore the ROIs and to select the one that really corresponds to the license plate.

---

\* This work has been supported by the Spanish *Ministerio de Ciencia y Tecnología* and FEDER under grant TIC2002-02684.



Fig. 1. Sample pictures of Spanish license plates in the database.

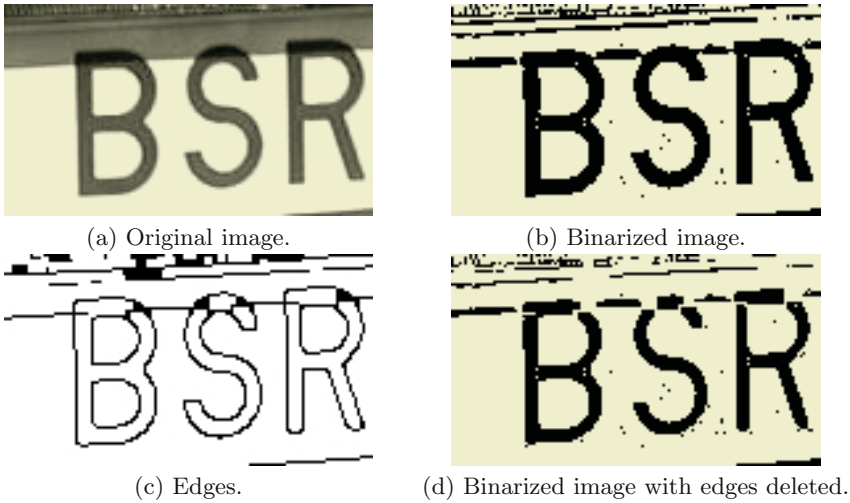
The paper is structured as follows: the next two sections explain the license plates extraction and recognition procedures, respectively; after that, we present experimental results of both stages and the whole system on a database of Spanish license plates; finally, we comment some conclusions and future work.

## 2 License Plates Extraction

The aim of this phase is to find a set of quadrangles that can be considered promising regions for holding a plate. Note that the plate cannot be assumed to have a rectangular position due to the perspective distortion introduced by the angle of the camera with respect to the car. For the same reason, the plate is not assumed to be horizontal. These regions (which we call ROI, for Regions Of Interest) are searched for by analyzing the connected components of a binarization of the image. This analysis looks for regions in which the components have certain properties such as being of similar height, having an aspect ratio in some range, and being roughly aligned.

The analysis of connected components is performed four times, each one using a different binarization. Similar plate candidates coming from different binarizations are combined and their scores are readjusted. Up to three candidates are selected by score. For each surviving plate candidate, a ROI is returned consisting of the minimum area quadrilateral fitting the bounding boxes of the connected components it contains.

In the following description we assume that the pictures are gray level images containing the frontal or rear side of a single vehicle. Plate width is assumed to



**Fig. 2.** Effect of removing edges from the binarized image: the characters of the plate are separated from one another.

be between 25% and 75% of the image width, which in our case is 800 pixels. Different sizes would need to change the values presented. No assumptions are made about the angle of the camera with respect to the vehicle. As mentioned above, this causes distortions in the image.

## 2.1 Binarization of the Image

Global thresholding is not appropriate in our case because the ROI usually is a small portion of the image and there can be large lighting differences in a scene. Therefore, we use a local thresholding technique: for each pixel, the threshold is computed by subtracting a constant  $c$  to the mean gray level in an  $n \times n$  window centered in the pixel. There is no single setting of  $n$  and  $c$  that has proven useful in all images of our training set. Thus, we use three different settings:  $(n = 20, c = 2)$ ,  $(n = 20, c = 6)$ , and  $(n = 9, c = 6)$ .

In some plates, shadows due to direct sunlight may link several characters (Figures 2 (a) and 2 (b)). To overcome this undesirable effect, a fourth binary image is produced by applying an edge detector to the original image (Figure 2 (c)) and removing the edges from the binary image obtained with parameters  $(n = 9, c = 6)$ . This is expected to disconnect the characters that were linked by the shadow (Figure 2 (d)).

## 2.2 Connected Component Analysis

Once the image is binarized, the process of analyzing the connected components for finding ROIs consists of the following steps: (a) connected components detection and filtering; (b) ROIs finding; and (c) ROIs scoring.

**Connected Components Detection and Filtering.** The binarized image is decomposed into 4-neighbours connected components. A filter removes small components (containing less than 100 pixels) and keeps those components whose width, height, and aspect ratio fall between some limits (25 and 140 for the height, 5 and 80 for the width, and 0.4 and 14 for the aspect ratio).

A possible problem with this filter is that it may miss some components corresponding to characters in the plate. However, note that this is only a problem when it affects to the leftmost and rightmost characters. Furthermore, we have seen experimentally that it is very unlikely that this happens on all four binarizations.

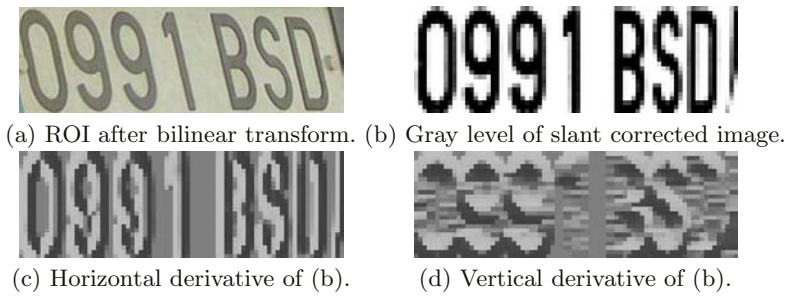
**ROIs Finding.** Once the connected components have been filtered the process of ROIs extraction begins. The aim of this process is to find sets of connected components containing at least four connected components of similar size and whose bounding box centers can be roughly fitted by a straight line. Only maximal sets are considered: *i.e.* if one such set is properly included in another, the first one is ignored. On the other hand, the sets need not be disjoint: a connected component can belong to more than one.

A simple analysis of the ROIs is performed in order to remove those components that lie too far away from the others. Figure 3 shows the ROIs found in the images of Figure 1.

**ROIs Scoring.** Each region of interest is scored attending to its number of connected components, overlapping of bounding boxes, and slope. The scores are defined so that higher scores are worst. The actual criteria are:



Fig. 3. ROIs found in the pictures of Figure 1.



**Fig. 4.** Preprocess and parameterization for the HMMs.

- The difference between the width of the candidate and the sum of the widths of the bounding boxes of the components is the basis score.
- If the number of components is below six or above eight, a penalty is added for each component of the difference with six (if it is below) and eight (if it is above).
- For each pair of components, the score is increased by the percentage of overlap between the corresponding bounding boxes.
- Finally, the score is increased with a multiple of the absolute value of slope of the line joining the centers of the components.

### 3 License Plates Recognition

We use Hidden Markov Models (HMM) as the basis model for our recognition engine. HMMs have been successfully used in speech recognition for a long time and more recently they have been applied to OCR tasks. The recognition begins with a preprocessing and a parameterization of the ROIs detected in the previous phase.

**Preprocessing.** The quadrilateral ROIs of the gray level image are mapped into rectangles by means of a bilinear transform (see Figure 4 (a)). These regions are supposed to contain only the plate, so a better binarization can be performed on them. We use a new local thresholding with a larger window. After binarization, a new connected component analysis removes noise. The slant of the surviving connected components is corrected and each component is rescaled to a standard height of 100 pixels.

**Parameterization.** We use a parameterization based on the one presented in [4]. The image is divided into a grid of  $20 \times N$  cells, where  $N$  is proportional to the width of the ROI. In each cell the average gray level and horizontal and vertical derivatives are computed. The gray level is a weighted average of the gray levels of a  $5 \times 5$  cells neighbourhood, the weights following a gaussian distribution (see Figure 4 (b)). The same neighbourhood is used in the computation of the derivatives. The horizontal derivative is defined as the slope of the line fitting

the average gray level in each row (see Figure 4 (c)). The vertical derivative is defined analogously (see Figure 4 (d)). With this process, the parameterization the ROI consists in  $N$  vectors of dimension  $3 \times 20$ .

**Hidden Markov Models.** An HMMs was trained for each character. Each model has a Bakis topology [3]: each state has three output arcs, one to itself, and one to each of the two following states. After some experiments performed on training images, the total number of states per character was fixed to 12, each one with a mixture of 128 Gaussian distributions. The models were trained on manually segmented images with the Hidden Markov Models Toolkit (HTK) [6].

**Language Model.** A standard practice in speech recognition [3] is to restrict the possible sequences by means of a *language model*. This is responsible for assigning an *a priori* probability to the different sequences. In our case, the valid Spanish license plates were encoded in a regular grammar. This is straightforward as there are currently two models of plate codes: (1) one or two letters that identify a province, four digits and one or two letters; (2) four digits and three consonants. In this first approach, all productions with the same left side were assigned the same probability.

## 4 Experiments

We have carried out some experiments with a corpus consisting of 468 images taken with a conventional digital camera (<ftp://acrata.act.uji.es/pub/MATRICES>). The images were resized to a standard width of 800 pixels. In the images, the width of the plate, after rescaling, lies between 25% and 75% of the width of the image (in pixels, between 200 and 600).

The images were divided in a group of 418 images for training and 50 images for test. The training images correspond to 341 vehicles (for some vehicles both the frontal and rear plates were taken) and the test images correspond to 43 vehicles. Care was taken to avoid the overlapping between the vehicles in the training and the test sets.

To ease the training procedures, the plates were transcribed and the bounding boxes of each character and plate were manually obtained.

### 4.1 Extraction Experiments

To evaluate the performance of the ROIs detection method, we have measured the number of times that the highest scored ROI matches the plate region on the test data. This happened in 45 of the 50 test plates. If the best three ROIS are considered, 49 plate regions are correctly identified. In all cases, the best ROI contained a significant part of the plate region. The result of the bilinear transforms of the best ROIs is presented in Figure 5.



Fig. 5. Highest scored ROIs in test pictures after the mapping to rectangles.

## 4.2 Recognition Experiments

HMMs parameters were estimated using a manual segmentation of the training images. In order to assess the HMMs, a first experiment was conducted on the manually segmented images of the test data. 94% of the plates were correctly recognized. A character accuracy rate of 98.1% was obtained. The errors were due to problems on overexposed and blurry images.

## 4.3 Global System Experiments

The ROIs obtained in the plates extraction stage were fed into the HMMs based recognizer. The average log-probability per column was used to rescore the ROIs and their associate license plate transcriptions. In this case, 88% of the plates were correctly recognized. In an additional 4% of the plates, a correct transcription was found for one ROI, but the combined score chose a wrong ROI, i.e. a better rescoring would have increased the recognition rate up to 92%. The ROI and HMMs scoring did not help to select the correct transcriptions in two misclassified plates, but the HMM recognizer correctly produced a transcription for their corresponding ROIs. The character accuracy rate was 95.7%.

## 5 Conclusions and Future Work

We have presented some preliminary results with a license plate extraction and recognition system based on connected components analysis and HMMs decoding. The results show the feasibility of this approach.

From the experimental results we can conclude that, although the ROI detector performance is very high, an improvement of the rescoring of ROIs is needed: the decrease in plate recognition from 94% (manual segmentation) to 88% is attributable to the fact that only in 90% of the cases the plate region corresponds to the first ROI. When all regions are taken into account, the recognition rate goes up to 92%, which corresponds to the 98% of times that plate region was one of the three best ROIs.

The HMMs produce near to 100% of correct characters on well parameterized regions of interest: the few decoding errors are due to extremely low quality images (severe overexposure) and inaccuracies in ROIs detection.

In the future we plan to improve the extraction by means of texture analysis such as the one proposed in [5]. Preliminary experiments with textures have shown that its results are not as good as the analysis of connected components, but it could be used to guide that analysis. We also plan to estimate the width of the strokes in the digits in order to improve the analysis of the connected components. This could be also employed to estimate the size of structural elements for applying morphological operators [1]. Another line of work is the recognition of plates in sequences of images from video streams, where movement information can be helpful to detect regions of interest.

## References

1. Antonio Albiol, J. Manuel Mossi, Alberto Albiol, and Valery Naranjo. Automatic license plate reading using mathematical morphology. In *Proceedings of the The 4th IASTED International Conference on Visualisation, Imaging and Image Processing*, Marbella, Spain, september 2004.
2. Fernando Martín, Maite García, and José Luis Alba. New methods for automatic reading of vlp's (vehicle license plates). In *SSPRA*, 2004.
3. Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77-2, pages 257–286, 1989.
4. Alejandro Héctor Toselli. *Reconocimiento de Texto Manuscrito Continuo*. PhD thesis, Departamento de Sistemas Informáticos y Computación, 2004.
5. Victor Wu, Raghavan Manmatha, and Edward M. Riseman. Textfinder: An automatic system to detect and recognize text in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1224–1229, 1999.
6. Steve Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Valtcho Valtchev, and Phil Woodland. *The HTK Book (for HTK Version 3.1)*. Cambridge University Engineering Dept., 2001. <http://htk.eng.cam.ac.uk/>.



# Multi-resolution Image Analysis for Vehicle Detection

Cristina Hilario, Juan Manuel Collado,  
José Maria Armingol, and Arturo de la Escalera

Grupo de Sistemas Inteligentes, Universidad Carlos III de Madrid  
C/ Butarque 15, 28911 Leganés, Spain  
{chilario, jcollado, armingol, escalera}@ing.uc3m.es

**Abstract.** Computer Vision can provide a great deal of assistance to Intelligent Vehicles. In this paper an Advanced Driver Assistance Systems for Vehicle Detection is presented. A geometric model of the vehicle is defined where its energy function includes information of the shape and symmetry of the vehicle and the shadow it produces. A genetic algorithm finds the optimum parameter values. As the algorithm receives information from a road detection module some geometric restrictions can be applied. A multi-resolution approach is used to speed up the algorithm and work in realtime. Examples of real images are shown to validate the algorithm.

## 1 Advanced Driver Assistance Systems

### 1.1 Motivation

Several Advanced Driver Assistance Systems (ADAS), that nowadays are being researched for Intelligent Vehicles, are based on Computer Vision [1]. One of them has the goal of detecting and tracking other vehicles. Present day, commercial equipments are based on distance sensors like radar or laser. These sensors have the advantage of giving a direct distance measurement and, above all, they are able to work under bad weather conditions. Their main inconvenience is the field of view, which is very narrow, so they can only detect the vehicle in front of the sensor. If the vehicle is overtaken, there is a step input to the system and the response can be unstable. One alternative or complementary sensor is vision. Although it is not able to work under bad weather conditions and its information is much difficult to process, it gives a richer description of the environment that surrounds the vehicle. Besides, many of the current traffic accidents happen under good weather and are due to human errors.

### 1.2 Previous Work

The research on vehicle detection based on an onboard computer vision system can be classified in three groups:

- Bottom-up. There are some features that define a vehicle (symmetry, edges, shadow), and they are looked for sequentially in the image. Their main inconveniences are: the vehicle is lost if one feature is not present enough in the image and false tracks can deceive the algorithm.

- Top-down. There are one or several models of vehicles and the best model is found in the image through a likelihood function. They are more robust than the previous algorithms, but slower. The algorithm presented in this article follows this approach.
- Learning based. Mainly, they are based on neural networks. Many images are needed to train the network. They are usually used in conjunction with a bottom-up algorithm to check if a vehicle has been actually detected. Otherwise, they have to scan the whole image and they are very slow

The shadow under the vehicles is looked for in [2]. To do so, a sample of the road just in front of the vehicle is taken and darker zones are searched. For these regions, symmetry and vertical edges confirm if there is a vehicle. A similar approach is found in [3]. In [4] a formula for symmetry is proposed. An elastic net is placed at the maximum and it is deformed until the vehicle is found. Interesting zones in the image are localized in [5] using Local Orientation Coding. A Back-propagation neural network confirms or rejects the presence of a vehicle. [6] follows the previous work but adding texture and shadows. The tracking is done using the Hausdorff distance to a model. Another example of fusing shadow, entropy and symmetry is found in [7]. In [8], shadows and symmetry are proposed to localize interesting zones; a neural network confirms the hypothesis. Symmetry is used in [9] to determine the column of the image where the vehicle is. After that, they look for an U-form pattern to find the vehicle. The tracking is performed with SSD correlation. They use a multi-resolution approach. Edges and symmetry are also used in [10]. In [11] overtaking vehicles are detected through motion (image difference) and the other vehicles through correlation. The dimension of the correlation window is calculated through edge detection. Several 3D models of vehicles are used in [12]. The road limits are calculated and the geometrical relationship between the camera and the road is known. Preceding vehicles are detected in [13]. They calculate a Multiclustered Modified Quadratic Discriminant Function through examples, and look for vehicles in regions of 16x16 pixels in the image.

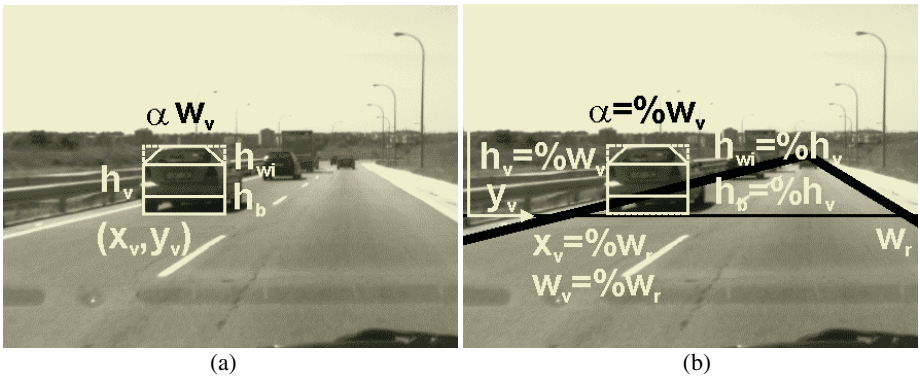
## 2 Geometrical Models

As stated in [14], a global shaped model based image segmentation scheme consists of the following blocks:

- The initial model,  $M$ .
- The deformable model  $M(Z)$ . This model is obtained from the previous one through the deformation parameters,  $Z$ .
- The likelihood probability density function,  $P(I|Z)$ , which means the probability of the deformation set  $Z$  occurs, in the image  $I$ .
- A search algorithm to find the maximum of the posterior probability  $P(Z|I)$ .
- The likelihood function  $P(I|Z)$  has to be designed to reach its maximum value when the deformed model matches image  $I$ .

### 2.1 Geometrical Model of a Vehicle

Due to shadows, occlusions, weather conditions, etc, the model has to incorporate as much information as possible. In this paper, a vehicle is defined by seven parameters



**Fig. 1.** Geometrical model of a vehicle. (a) A vehicle is defined by seven parameters: Position (x,y), width and height of the vehicle, windshield position, bumper position and roof angle (b) The values of this parameters are constrained by the detection of the road.

(Fig. 1-a): Position (x,y), width and height of the vehicle, windshield position, bumper position and roof angle. In a previous research, [15], the seven parameters had a range but, while the range of the X and Y position, and the width and height of the vehicle were in pixels, the range of the windshield and bumper position and the roof angle were a percentage of the height or width.

A previous detection of the road limits is done in [2] [10]. This can help the vehicle detection step because the searched area is smaller. In the present case, both borders of the road are found and modelled by equations:

$$x = f_l(y) \quad x = f_r(y) \tag{1}$$

that are the slope of the straight lines in this case, but the algorithm would be the same if they were parabolas or clotoids. For a specific  $y_v$  value (Fig. 1-b), the width of the road is found:

$$x_l = f_l(y_v) \quad x_r = f_r(y_v) \Rightarrow w_r = x_r - x_l \tag{2}$$

The  $x_v$  value of the vehicle and its width are two percentages of the width of the road:

$$x_v = K_x w_r \quad w_v = K_w w_r \tag{3}$$

The height of the vehicle is proportional to the width:

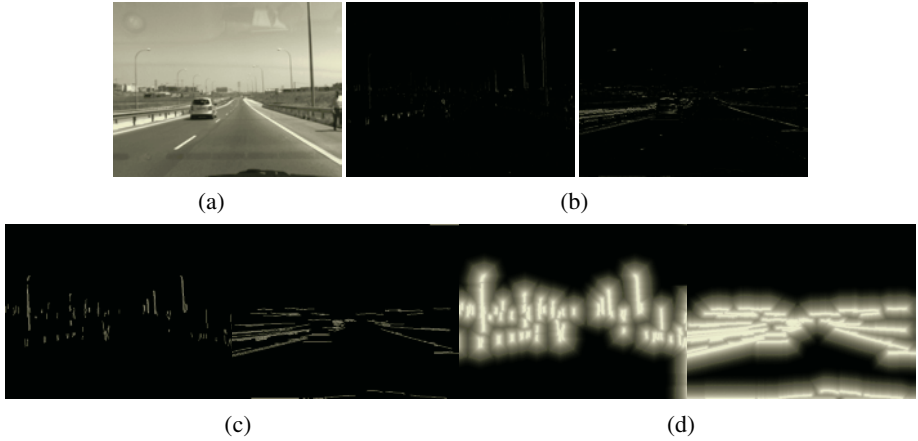
$$h_v = K_h w_v \tag{4}$$

And finally, the windshield and bumper position and the roof angle are a percentage of the height or width.

$$h_{wi} = K_{wi} h_v \quad h_b = K_b h_v \quad \alpha = K_\alpha w_v \tag{5}$$

Then, the deformation parameter vector is:

$$Z = \{y_v, K_x, K_w, K_h, K_{wi}, K_b, K_\alpha\} \tag{6}$$



**Fig. 2.** Image processing (a) Image (b) Vertical and horizontal gradients (c) Vertical and horizontal edges (d) vertical and horizontal distances

### 2.2 Energy Function

The energy function considers the following three factors: Symmetry, shape and the vehicle shadow (Fig. 2).

#### Symmetry

The symmetry of the vertical and horizontal edges is considered. For this reason, the vertical and horizontal gradient components of the image are found (Fig. 2-b, Fig. 2-c). Only the pixels with a high response in one of the components and low in the other are taken into account. Then, the pairs of pixels in the same line vote for the central pixel as their symmetry axe. The formulae can be found in [15].

#### Shape

Shape is defined by two energy terms: one based on the gradient (Fig. 2-b) and the other one based on the distance to the edges, found before for the symmetry energy (Fig. 2-d). The formulae can be found in [15]. Here, only the distance formula is explained, because it has changed from the previous research. A distance image is obtained where each pixel shows the distance to the nearest edge. In order to emphasized the pixels that are near to the edges, the following look up table is applied

$$Lut(D) = \begin{cases} 255(1 - \text{sqrt}(D / D_{\max})) & 0 < D < D_{\max} \\ 0 & D > D_{\max} \end{cases} \quad (7)$$

From that image, a distance to vertical edge energy,  $D_{GV}$ , and horizontal edge energy,  $D_{GH}$ , are calculated, where  $D_G$  is the global distance energy.

$$D_{GV} = \frac{1}{2h} \left( \sum_{j=y}^{y+h} Dv(x, j) + \sum_{j=y}^{y+h} Dv(x + w, j) \right) \quad (8)$$

$$D_{GH} = \frac{1}{4w} \left( \sum_{i=x}^{x+w} Dh(i, y) + \sum_{i=x}^{x+w} Dh(i, y+t) + \sum_{i=x}^{x+w} Dh(i, y+m) + \sum_{i=x}^{x+w} Dh(i, y+h) \right) \quad (9)$$

$$D_G = \frac{(D_{GV} + D_{GH})}{2}. \quad (10)$$

### Shadow

The shadow energy,  $E_{SOM}$ , of a vehicle with height  $h$ , width  $w$ , position  $(x,y)$ , and bumper position  $m$ , is defined by the average level of grey in the lower part of the model. Again, the formulae can be found in [15].

### Global Energy

The final energy,  $E$ , is:

$$E(Z) = -(k_A E_{Sim}(Z) + k_B E_G(Z) + k_C E_D(Z) + k_D E_{Som}(Z)) \quad (11)$$

where  $k_A$ ,  $k_B$ ,  $k_C$  and  $k_D$  allow a weighted sum of the energy terms.

## 2.3 Likelihood Probability Density Function

The estimate of a given deformation  $Z$  for the image  $I$ ,  $P(I|Z)$ , follows a Gibbs distribution [14]:

$$P(I | Z) = \frac{1}{K} \exp - E(Z) \quad (12)$$

where  $K$  is the normalizing constant.

The detection problem is the search of the Maximum A Posteriori (MAP) estimation of  $Z$ .

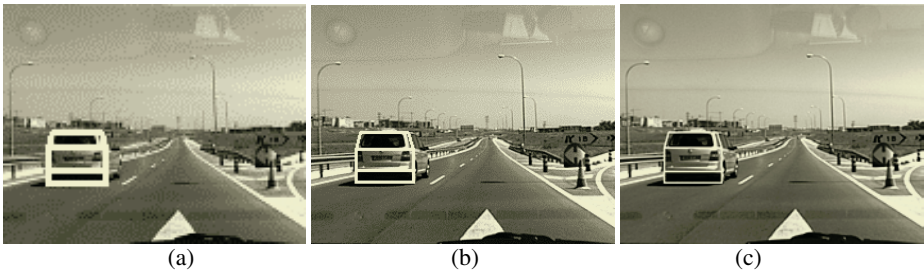
$$Z_{MAP} \in \arg \max_Z P(I | Z) \in \arg \min_Z E(Z) \quad (13)$$

The energy function is minimal when the deformed model exactly matches with the one presented in the image.

## 2.4 Search Algorithm

Search algorithms have to find a balance between two opposite tasks: exploration of the complete search space and the exploitation of certain zones. With exploration, the search space is covered looking for promising areas in which a more detailed search has to be done; that is the exploitation task, where the best solution is looked for in a zone known as suitable. The risk is being trapped in a local maximum or minimum. Hashing methods are the extreme case of exploration, where gradient-based methods (hillclimbing) are the extreme for exploitation.

Genetic algorithms (GAs) [16] do a parallel search in several directions following an optimisation process, which imitates natural selection and evolution. To accomplish this task, there is a set of possible solutions (the individuals) that exchange information depending on the fitness of the result in the search for the global maximum. GAs robustness relies in their ability to reach a global maximum surrounded by local ones.



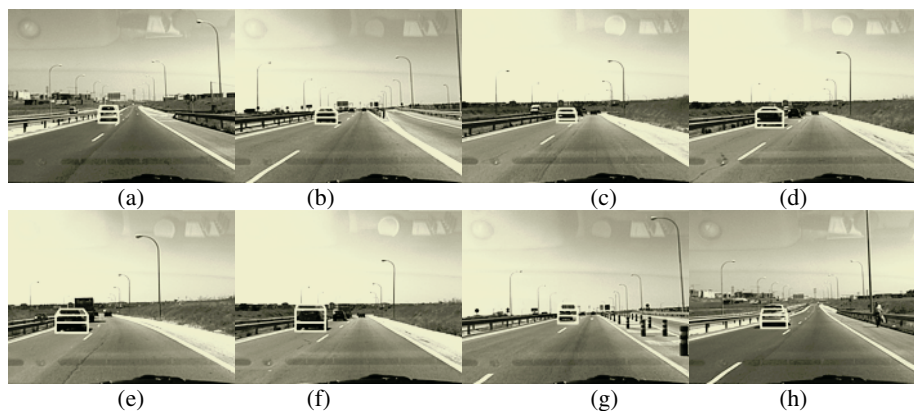
**Fig. 3.** Multi-resolution detection at (a) 160x120 (b) 320x240 (c) 640x480 pixels



**Fig. 4.** Advantages of a multi-resolution approach. (a) small errors in the vehicle detection (b) multi-resolution detection (c) wrong detection (d) multi-resolution detection

### 3 Results

The detection of the vehicle is done for multiple resolutions. A Gaussian pyramid is built, with dimensions: 160x120, 320x240, and 640x480 pixels. The information of the detection of lower levels is passed to greater levels (Fig.3). Working with a multi-resolution approach has the main advantage of working with the best resolution for every circumstance. Take for example Fig. 4-a. The vehicle has been detected but, as there are many edges inside the car, there are some small errors in the detection. Those edges inside the car have less importance at a lower resolution and the detection is better (Fig. 4-b). But, not only it is useful to improve the results but also to detect successfully a vehicle. As the vehicle is in a cluttered environment, some edges in the environment can deceive the algorithm if an image with great detail is used (Fig. 4-c). Again, working first with a smaller image improves the results (Fig. 4-d). Another advantage is the saving in computational time. In [15] 550 individuals were needed to detect the vehicles in front of the camera. With the present approach, only 32 individuals are needed. That means the algorithm spends now an average time for a genetic generation of 0.16 ms instead of the 24ms of [15] (in a Pentium 4 Mobile at 1.7 GHz). The other parameters of the GA algorithm are:



**Fig. 5.** Some results and errors. (a)-(e) Successful detection of vehicles. (f)(g) Other rectangular objects are taken as part of the vehicle (h) an inner part of the vehicle is taken.

- Crossover probability: 70%
- Mutation probability: 3%
- Elitism

More results are shown, from Fig. 5-a to Fig. 5-e. Some errors are also shown. In Fig. 5-f-g, the vehicle detected is taller than the real one. This is because some rectangular objects in the environment, like buildings or informative signs are taking as part of the vehicle. Also, when the vehicle is very close to the camera, a smaller vehicle is detected (Fig. 5-h).

## 4 Conclusions

A system based on computer vision for the detection of other vehicles has been presented in this paper. It is based on a geometric model and its energy function includes information of the shape and symmetry of the vehicle and the shadow it produces. A genetic algorithm has been used to find the optimum parameter values. The algorithm is able to detect vehicles in front of the camera, and it can also detect lateral vehicles and trucks.

## Acknowledgments

This work was supported in part by the Spanish CICYT Grant TRA2004-07441-C03-01.

## References

1. Dickmanns, E.D.: The development of machine vision for road vehicles in the last decade. IEEE Intelligent Vehicles Symposium (2002) 268-281
2. Charkari, N. M., Mori H.: Visual vehicle detection and tracking based on the sign pattern. Advanced Robotics 9 (1995) 367-382

3. Hoffmann, C., Dang, T., Stiller, C.: Vehicle detection fusing 2D visual features. *IEEE Intelligent Vehicles Symposium* (2004) 280-285
4. Zielke, T., Brauckmann M., Von Seelen, W.: Intensity and edge-based symmetry detection with application to car-following. *CVGIP: Image Understanding* 58 (1993) 177-190
5. Goerick, C., Noll, D., Werner, M.: Artificial Neural Networks in Real Time Car detection and Tracking Applications. *Pattern recognition Letters*, 17, (1996) 335-343
6. Handmann, U., Kalinke, T., Tzomakas, C., Werner, M., Goerick C., von Seelen, W.: An image processing system for driver assistance. *Image and Vision Computing* 18 (2000) 367-376
7. ten Kate, T.K., van Leewen, M.B., Moro-Ellenberger, S.E., Driessen, B.J.F., Versluis, A.H.G., Groen, F.C.A.: Mid-range and Distant Vehicle Detection with a Mobile Camera. *IEEE Intelligent Vehicles Symposium* (2004) 72-77
8. Matthews, N.D., An, P.E., Roberts, J.M., Harris, C.J.: A neurofuzzy approach to future intelligent driver support systems. *Proceedings-of the Institution of Mechanical Engineers Part D (Journal of Automobile Engineering)* 212 (1998) 43-58
9. Broggi, A., Cerri, P., Antonello, P.C.: Multi-Resolution Vehicle Detection using Artificial Vision. *IEEE Intelligent Vehicles Symposium* (2004) 310-314
10. Sotelo, M.A., Fernandez, D., Naranjo, J.E., González, C., García, R., de Pedro, T., Reviejo, J.: Vision-based Adaptive Cruise Control for Intelligent Road Vehicles. *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2004) 64-69
11. Betke, M., Haritaoglu, E., Davis, L.S.: Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications* 12 (2000) 69-83
12. Ferryman, J.M., Maybank S.J., Worrall, A.D.: Visual surveillance for moving vehicles. *International Journal of Computer-Vision* 37 (2000) 187-97
13. Kato, T., Ninomiya Y., Masaki I.: Preceding vehicle recognition based on learning from sample images, *IEEE Transactions on Intelligent Transportation Systems* 3 (2002) 252-260
14. Dubuisson, M-P., Lakshmanan S., Jain A.K.: Vehicle segmentation and classification using deformable templates. *IEEE Transactions on Pattern analysis and Machine Intelligence* 18 (1998) 293-308
15. Collado, J. M., Hilario, C., de la Escalera, A., Armingol, J. M<sup>a</sup>: Model Based Vehicle Detection for Intelligent Vehicles. *IEEE Intelligent Vehicles Symposium* (2004) 572-577
16. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Boston (1989)



# A Novel Adaptive Gaussian Mixture Model for Background Subtraction

Jian Cheng, Jie Yang, and Yue Zhou

Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University  
Shanghai 200030, China  
{ch\_jian, jieyang, zhoyue}@sjtu.edu.cn

**Abstract.** Background subtraction is a typical approach to foreground segmentation by comparing each new frame with a learned model of the scene background in image sequences taken from a static camera. In this paper, we propose a flexible method to estimate the background model with the finite Gaussian mixture model. A stochastic approximation procedure is used to recursively estimate the parameters of the Gaussian mixture model, and to simultaneously obtain the asymptotically optimal number of the mixture components. The experimental results show our method is efficient and effective.

## 1 Introduction

Background subtraction is a typical approach to segment moving object by comparing each new frame with a learned model of the scene background in image sequences taken from a static camera. Many researchers have made previous attempts to segment moving object by background subtraction [1,2,3,4]. An effective and adaptive approach to background subtract is to construct a statistical model which represents the probabilistic distribution of the pixel's intensity or color. Wren et al. adopt a single Gaussian to represent the background model [1]. However, this system is sensitive to the initialization, and is improper to process multi-modal and clutter scenes. Another statistical model for background subtraction is the finite Gaussian mixture model (GMM) [2,3,4]. Friedman and Russell [2] use a mixture of three Gaussian distributions to model the pixel value for traffic surveillance applications. Stauffer et al. [3,4] propose a similar algorithm, which uses a mixture of Gaussian distribution to model a multi-modal background. However, these methods all have a drawback that the number of the mixture components is a pre-set and fixed value. Because the number of the mixture components mostly determines the number of the need-estimating parameters, this drawback may make foreground segmentation time-consuming.

Obviously, the Gaussian mixture model is an effective approach to background subtraction for the multi-modal and clutter scene. However, learning the GMM parameters is computationally expensive, and an efficient learning algorithm is the key to GMM for background subtraction. The previous researchers mostly adopt the Expectation Maximization (EM) algorithm [5] to learn the GMM parameters. However, A serious drawback of the EM algorithm is that it can converge to a poor local maximum if not properly initialized [6,7,8]. Moreover, there are two important problems

when GMM is used to model multivariate data: the selection of the number of components and the initialization. Figueiredo and Jain propose an unsupervised algorithm for learning a finite mixture model from multivariate data [7]. Their algorithm has two properties: 1) automatically selects the number of components, 2) is less sensitive to initialize. Recently, Zivkovic and Heijden [8] develop Figueiredo and Jain's research, and propose an online algorithm that estimates the parameters of the mixture model and simultaneously selects the number of components. In this paper, we propose a flexible method for background subtraction. We still use the finite Gaussian mixture model to model the scene background, but a stochastic approximation procedure is used to recursively estimate the parameters of the Gaussian mixture model, and to simultaneously obtain the asymptotically optimal number of the mixture components like [8]. Therefore, our method is highly memory and time efficient. Moreover, our method can effectively deal with the outdoor scene and the clutter scene.

## 2 Flexible Gaussian Mixture Models for Background Subtraction

In general, the pixel intensity is modeled by a mixture of  $K$  Gaussian distributions to model significant variations in the background.  $K$  is the component number of the mixture corresponding to  $K$  modes of the background, and is a fixed number from 3 to 7 in the most existing papers. However, the mode of most pixels in the scene background is different, for example, in the same outdoor scene the mode number of the pixels that contain tree branches and bushes movement by wind is larger than one of the pixels that don't contain. Clearly, it is unreasonable to specify a fixed number mixture model to represent the scene background. Fortunately, the component number of the mixture model can be on-line learned as [8]. In our method, we still adopt the Gaussian mixture model to model the scene background. However, for each pixel in the background, the component number of the Gaussian mixture model is not fixed, obtained by on-line estimation as the parameters of the Gaussian mixture model.

### 2.1 Flexible GMM for Background Estimation

Assume that  $X = \{\bar{x}_1, \dots, \bar{x}_t\}$  is a pixel value process, which can be modeled by a mixture of  $K$  Gaussian densities with  $\theta = [w_1, \dots, w_K; \bar{\mu}_1, \dots, \bar{\mu}_K; \Sigma_1, \dots, \Sigma_K]$ . The probability of the current pixel value  $\bar{x}_t$  is

$$p(\bar{x}_t | \theta) = \sum_{k=1}^K \frac{w_k}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\bar{x}_t - \bar{\mu}_k)^T \Sigma_k^{-1} (\bar{x}_t - \bar{\mu}_k)}, \quad (1)$$

where  $\bar{x}_t$  is the color or intensity value of the pixel,  $w_k$  is the weight of the  $k$ th Gaussian in the mixture,  $\bar{\mu}_k$  is the mean value of the  $k$ th Gaussian in the mixture,  $\Sigma_k$  is the covariance of the  $k$ th Gaussian in the mixture. For simplification, we assume

that the red, green, and blue color channels are independent and have a same variance, then  $\Sigma_k = \sigma_k I$

Given by Bayes's rule, the posterior probability  $p(\hat{\theta}(K)|X)$  is

$$p(\hat{\theta}(K)|X) = \frac{p(X|\hat{\theta}(K))p(\hat{\theta}(K))}{p(X)}. \tag{2}$$

Then, the MAP estimation can be obtained by maximizing  $\log p(\hat{\theta}(K)|X)$ .

Moreover,  $p(X)$  is independent of  $K$  in (2). Then, the MAP estimation is

$$\begin{aligned} \hat{\theta} &= \arg \max_{\hat{\theta}} \left( \log p(\hat{\theta}(K)|X) \right) \\ &= \arg \max_{\hat{\theta}} \left( \log p(X|\hat{\theta}(K)) + \log p(\hat{\theta}(K)) \right). \end{aligned} \tag{3}$$

As in [7] and [8], we define  $p(\hat{\theta}(K))$  as the Dirichlet prior

$$p(\hat{\theta}(K)) \propto \exp \sum_{k=1}^K -\alpha \log w_k = \prod_{k=1}^K w_k^{-\alpha}, \tag{4}$$

where  $\alpha$  equals to  $N/2$ , and  $N$  is the number of parameters per component.

### 2.2 Online Parameters Estimation

For the MAP estimation, let  $\frac{\partial}{\partial \hat{\theta}} \left( \log \left( p(X|\hat{\theta}) \right) + \log \left( p(\hat{\theta}) \right) \right) = 0$ , where

$p(\hat{\theta})$  is the previously defined Dirichlet prior as (4). Moreover, the mixture weights are constrained to sum up to 1. By introducing the Lagrange factor  $\lambda$ , we can get

$$\frac{\partial}{\partial \hat{w}_k} \left( \log \left( p(X|\hat{\theta}) \right) + \log \left( p(\hat{\theta}) \right) + \lambda \left( 1 - \sum_{k=1}^K \hat{w}_k \right) \right) = 0. \tag{5}$$

After getting rid of  $\lambda$ , for  $t$  data samples, we will get

$$\hat{w}_k^{(t)} = \frac{1}{C} \left( \sum_{i=1}^t o_k^{(i)}(\bar{x}^{(i)}) - \alpha \right), \tag{6}$$

where  $o_k^{(t)}(\bar{x}) = \hat{w}_k^{(t)} p_k(\bar{x} | \hat{\theta}_k^{(t)}) / p(\bar{x} | \hat{\theta}^{(t)})$  is the ‘ownership’ function that has a value from 0 to 1, indicating which class the sample belongs to,  $C = \sum_{k=1}^K \left( \sum_{i=1}^t o_k^{(t)}(\bar{x}^{(i)}) - \alpha \right) = t - K\alpha$  since  $\sum_{i=1}^t o_k^{(t)}(\bar{x}^{(i)}) = 1$ . Then, (6) can be rewrite

$$\hat{w}_k^{(t)} = \frac{\hat{\psi}_k - \alpha/t}{1 - K\alpha/t}, \tag{7}$$

where  $\hat{\psi}_k = \frac{1}{t} \sum_{i=1}^t o_k^{(t)}(\bar{x}^{(i)})$  and the bias from the prior is introduced by  $\alpha/t$ . The bias decreases for large data sets. If a small bias is acceptable, we can keep is constant by fixing  $\alpha/t$  to  $\alpha_T = \alpha/T$  with some large T. Then, as [8] we can get the recursive update equation

$$\hat{w}_k^{(t+1)} = \hat{w}_k^{(t)} + (1+t)^{-1} \left( \frac{o_k^{(t)}(\bar{x}^{(t+1)})}{1 - K\alpha_T} - \hat{w}_k^{(t)} \right) - (1+t)^{-1} \frac{\alpha_T}{1 - K\alpha_T}, \tag{8}$$

where T should be sufficiently large to make sure that  $K\alpha_T < 1$ . We start with initial  $\hat{w}_k^{(0)} = 1/K$  and discard the  $k$ th component when  $\hat{w}_k^{(n+1)} < 0$ . While the recursive equations of its mean  $\bar{\mu}_k$  and covariance matrix  $\hat{\Sigma}_k$  is

$$\hat{\mu}_k^{(t+1)} = \hat{\mu}_k^{(t)} + (t+1)^{-1} \frac{o_k^{(t)}(\bar{x}^{(t+1)})}{\hat{w}_k^{(t)}} (\bar{x}^{(t+1)} - \hat{\mu}_k^{(t)}) \tag{9}$$

$$\hat{\Sigma}_k^{(t+1)} = \hat{\Sigma}_k^{(t)} + (t+1)^{-1} \frac{o_k^{(t)}(\bar{x}^{(t+1)})}{\hat{w}_k^{(t)}} \left( (\bar{x}^{(t+1)} - \hat{\mu}_k^{(t)}) (\bar{x}^{(t+1)} - \hat{\mu}_k^{(t)})^T - \hat{\Sigma}_k^{(t)} \right). \tag{10}$$

### 2.3 Background Subtraction

The Gaussian mixture model with  $K$  components as (1) models both the foreground object and the scene background without distinction, that is, some of the mixture components model the foreground objects, others model the scene background. If one mixture component occurs frequently (with high  $w_k$ ), and does not vary much (with low  $\sigma_k$ ), it could be deemed to be background. Therefore, the  $K$  mixture components are ordered based on  $w_k / \sigma_k$ , moreover, the first  $B$  components are chosen as a model of the scene background where  $B$  is estimated as

$$B = \arg \min_b \left( \frac{\sum_{i=1}^b w_i}{\sum_{k=1}^K w_k} > T \right). \tag{11}$$

$T$  is the threshold, the fraction of the total weight given to the background model. Background subtraction is performed, by marking any pixel of the input frame that is more than 2.0 standard deviations away from any of the  $B$  components as a foreground.

### 2.4 A Practical Algorithm

In general, a practical algorithm for foreground segmentation with the Gaussian mixture model comprises four steps: Initialization, Background learning, Background subtraction, and Background update.

1) Algorithm Initialization: In order to make the algorithm time and memory efficient, we start with a proper number of components  $K$ . In our experiments, the initial component value is  $K = 1$ . Then, the first frame can be immediately used to initialize the mean  $\hat{\mu}_k^{(0)}$ . The initial covariance is a relatively high value, such as  $\hat{\sigma}_k^{(0)} = 0.2$ .

Finally, the initial mixing weights are set as  $\hat{w}_k^{(0)} = 1/K$ .

2) Background model learning: For simplification, it is reasonable to fix the influence of the new samples by replacing the term  $(1+t)^{-1}$  from the recursive equations (8), (9), and (10) by  $\beta = 1/T$ . A fixed  $\beta$  can help in forgetting the out-of-date statistics more rapidly. In addition, it is reasonable to let  $1 - K\alpha_T \doteq 1$ , since  $K$  is a small value in our experiments. Then, the equation (8) will be rewritten

$$\hat{w}_k^{(t+1)} = \hat{w}_k^{(t)} + \beta \left( o_k^{(t)} \left( \bar{x}^{(t+1)} \right) - \hat{w}_k^{(t)} \right) - \beta \alpha_T. \tag{12}$$

Besides, in order to adaptively estimate the number of the mixture components, we define two rules, which are the generation rule and the extinction rule.

- Generation rule: if not ‘match’ any mixture components, generate a new component, with  $K = K + 1$ , and the initial parameters are  $w_{K+1}^{(t+1)} = \lambda$ ,  $\mu_{K+1}^{(t+1)} = \bar{x}^{(t+1)}$ ,  $\sigma_{K+1}^{(t+1)} = 0.2$ , where  $\lambda$  is a very small fraction.
- Extinction rule: if  $\hat{w}_k^{(t+1)} < 0$ , discard this component  $k$ , with  $K = K - 1$ .

3) Background subtraction: As what section 2.3 discussed, we can extract the foreground object by background subtraction.

4) Background model update: Background model update is same as background model learning. Resorting to the online background model update, background model can adapt the scene background.

### 3 Experimental Results

Two experiments are exhibited here, which demonstrate the algorithm performance on the outdoor scene and the clutter scene. The image size of all test sequences is  $160 \times 120$  pixels. These experiments are implemented on the MATLAB 6.5 platform with the Pentium4 2.4GHz and 512MB memory, and the processing-ratio of two experiments is approximate to 4 fps (frames per second). The experimental results show that our method is efficient and effective.

The first experiment is the outdoor scene. Many outdoor scenes contain not only the moving foreground objects, but also the moving background, such as tree branches and bushes swaying by wind. In order to model the moving background, our method uses the mixture of multiple Gaussian components, while the number of the mixture component is online estimate, and is asymptotically optimal. In the test sequence, a person walks in front a swaying tree. Fig. 1 shows the foreground segmentation result. In second experiment, the sequence is captured in a cafeteria with many people. Moreover, every frame contains moving people. For example, the 420th frame contains 3 moving people, and the 450th frame contains 8 moving people. Similarly, in order to model the clutter scene, our method uses the asymptotically optimal mixture of multiple Gaussian components. Fig. 2 shows the foreground segmentation results.

The two experimental results show our proposed method is effective for the outdoor scene and the clutter scene. Moreover, our method is also efficient. In our experiments, we find that the mixture component number, which is used to model the different pixels in the scene background, may be different. One part of background pixels, which are relatively changeless and static, can be represented only by a single Gaussian component, while the other part of background pixels, which are changeful and non-static, must be modeled by the mixture of multiple Gaussian components.



**Fig. 1.** Foreground segmentation in the outdoor scene with tree branches and bushes swaying, (a) is the original image of the 251st frame, (b) is the segmentation result.



**Fig. 2.** Foreground segmentation in the clutter scene, (a) and (c) are the original images of the 420th and 450th frame, (b) and (d) are the segmentation result.

## 4 Conclusions

A flexible Gaussian mixture model for background subtraction is proposed. The model can handle the outdoor scene where the background is not completely static but contains tree branches and bushes motion, and the clutter scene with multiple foreground objects. Moreover, the number of the mixture component is not a pre-set and fixed value, but it is online estimated, and is asymptotically optimal. Therefore, our method is more time and memory efficient than the Gaussian mixture model with the fixed component number.

## References

1. C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.19, No.7: 780-785, 1997.
2. N. Friedman, S. Russell. Image Segmentation in Video Sequences: A Probabilistic Approach. in *Proc. of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, Aug. 1997.
3. W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using Adaptive Tracking to Classify and Monitor Activities in a Site. in *Computer Vision and Pattern Recognition*, 1998.
4. C. Stauffer and W.E.L. Grimson. Adaptive Background Mixture Models for Real-Time Tracking. in *Computer Vision and Pattern Recognition*, 1999.
5. A.P. Dempster, N. Laird, and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Royal Statistical Soc., Series B*, Vol. 1, No. 39:1-38, 1977.
6. M.E. Brand. Structure Learning in Conditional Probability Models via an Entropic Prior and Parameter Extinction. *Neural Computation J*, Vol. 11, No. 5: 1155-1182, 1999.
7. M. Figueiredo and A.K. Jain. Unsupervised Learning of Finite Mixture Models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 24, No. 3:381-396, 2002.
8. Z. Zivkovic and F. van der Heijden. Recursive Unsupervised Learning of Finite Mixture Models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 26, No. 5: 651-656, 2004.

# Intelligent Target Recognition Based on Wavelet Adaptive Network Based Fuzzy Inference System

Engin Avci<sup>1</sup>, Ibrahim Turkoglu<sup>1</sup>, and Mustafa Poyraz<sup>2</sup>

<sup>1</sup> Firat University, Department of Electronic and Computer Science  
23119, Elazig, Turkey  
enginavci@firat.edu.tr

<sup>2</sup> Firat University, Engineering Faculty, Department of Electric and Electronic  
23119, Elazig, Turkey  
mpoyraz@firat.edu.tr

**Abstract.** In this paper, an intelligent target recognition system is presented for target recognition from target echo signal of High Resolution Range (HRR) radars. This paper especially deals with combination of the feature extraction and classification from measured real target echo signal waveforms using X –band pulse radar. Because of this, a wavelet adaptive network based fuzzy inference system model developed by us is used. The model consists of two layers: wavelet and adaptive network based fuzzy inference system. The wavelet layer is used for adaptive feature extraction in the time-frequency domain and is composed of wavelet decomposition and wavelet entropy. The used for classification is an adaptive network based fuzzy inference system. The performance of the developed system has been evaluated in noisy radar target echo signals. The test results showed that this system was effective in detecting real radar target echo signals. The correct classification rate was about 93% for used target subjects.

**Keywords:** Pattern recognition, Radar Target Echo Signal, Feature extraction, Wavelet decomposition, Entropy, Wavelet adaptive network based fuzzy inference system, Intelligent system.

## 1 Introduction

This study will introduce the technique that will aid automatic target recognition, enable further research of target recognition, and provide a novel intelligent system for target recognition [1-3]. This study uses a combination of wavelet signal processing and adaptive network based fuzzy inference system to efficiently extract the features from pre-processed real target echo signals for the purpose of automatic target recognition among variety targets. An algorithm called the intelligent system is developed which is advanced pattern recognition approximation.

In radar automatic target recognition and multiple-target tracking areas, the novelties presented of this paper can be summarized as follow:

1. The presented first novelty in this study is using an effectively adaptive feature extraction method that increases percent of the target recognition.
2. The presented second novelty in this study is using of the wavelet adaptive network based fuzzy inference system as a new and efficiently method in radar automatic target recognition area.



In this study, an experiment set is used for obtaining the real target echo signal data sets. The Radar experiment set is educational purpose and multi function 9620/21 Model Lab-Volt radar experiment set. Pulse target echo signals are received to computer media by using an audio card has 44 KHz sample frequencies.

## 2 Preliminaries

### 2.1 Target Echo Signals

The echo signal comes back from target to Radar. At the same time, the echo signal can be called target range profile. In literature, there are many studies, in which echo signal were used for automatic target recognition [1-5].

In this study, pulsed radar target echo signals were used as real input space. An efficiency feature extraction method was developed for eight target objects (small metal plaque, large metal plaque, large plexiglas plaque, corner reflector, sphere, the side part of cylinder, the lower part of cylinder, and the crosswise part of cylinder) to separate one from the others. Experimental application was realized on having educational purpose and multi function 9620/21 Model Lab-Volt radar experiment set. Pulse echo signals were received to computer media by using audio card has 44 KHz sample frequencies.

### 2.2 Wavelet Decomposition

Wavelet transforms are rapidly surfacing in fields as diverse as telecommunications and radar target recognition. Because of their suitability for analyzing non-stationary signals, they have become a powerful alternative to Fourier methods in many target recognition applications, where such signals abound [6].

The main advantages of wavelets is that they have a varying window size, being wide for slow frequencies and narrow for the fast ones, thus leading to an optimal time-frequency resolution in all frequency ranges. Furthermore, owing to the fact that windows are adapted to the transients of each scale, wavelets lack of the requirement of stationary [7].

The wavelet decomposition functions at level  $m$  and time location  $t_m$  can be expressed as Equation 1:

$$d_m(t_m) = x(t)\Psi_m\left(\frac{t-t_m}{2^m}\right) \quad (1)$$

where  $\Psi_m$  is the decomposition filter at frequency level  $m$ . The effect of the decomposition filter is scaled by the factor  $2^m$  at stage  $m$ , but otherwise the shape is the same at all stages [8].

### 2.3 Wavelet Adaptive Network Based Fuzzy Inference System

Both artificial neural network and fuzzy logic are used in ANFIS's architecture. ANFIS is consisted of if-then rules and couples of input-output, for ANFIS training is used learning algorithms of neural network [10].

Adaptive network based fuzzy inference systems are good at tasks such as pattern matching and classification, function approximation, optimization and data clustering, while traditional computers, because of their architecture, are inefficient at these tasks, especially pattern-matching tasks [11,12]. As for wavelet adaptive network based fuzzy inference system try to combine aspects of the wavelet transformation for purpose of feature extraction and selection with the characteristic decision capabilities of adaptive network based fuzzy inference system approaches [13]. The wavelet adaptive network based fuzzy inference system (WANFIS) is constructed based on the wavelet transform theory [14,15] and is an alternative to adaptive network based fuzzy inference systems [16]. Wavelet decomposition [9] is a powerful tool for non-stationary signal analysis. Let  $x(t)$  be a piecewise continuous function. Wavelet decomposition allows one to decompose  $x(t)$  using a wavelet function  $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$ . Based on the wavelet decomposition, wavelet adaptive network based fuzzy inference system structure is defined by

$$y(x) = \frac{\sum_i w_i f_i(\psi[D_i(x - t_i)])}{\sum_i w_i} \quad (2)$$

where  $w_i$  are weights of the WANFIS inputs,  $D_i$  are dilation vectors specifying the diagonal dilation matrices  $D_i$ ,  $t_i$  are translation vectors, and  $f_i$  are Sugeno output functions of the ANFIS. An algorithm of the hybrid type has been derived for adjusting the parameters of the WANFIS [10-12]. Applications of wavelet adaptive network based fuzzy inference system in the medical field include for detection of electrocardiography changes in patients with partial epilepsy using feature extraction [16], bearing fault diagnosis based on wavelet transform and fuzzy inference [17], for satellite image fusion [18]; however, to date wavelet adaptive network based fuzzy inference system analysis of radar target echo signal is a relatively new approach.

### 3 Methodology

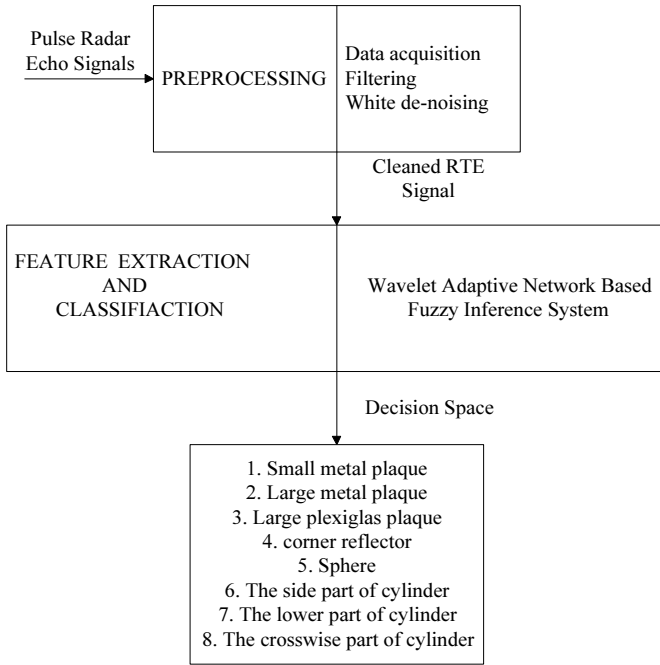
Fig. 1 shows the intelligent system we developed. It consists of two parts: (a) data acquisition and pre-processing and (b) feature extraction and classification using a wavelet adaptive network based fuzzy inference system.

#### 3.1 Data Acquisition and Pre-processing

All the original Radar Target Echo (RTE) signals were acquired from the having educational purpose and multi function 9620/21 Model Lab-Volt radar experimental set. The pulsed radar system parameters were adjusted as below:

- Pulse width: 2 ns
- RF oscillator: 9.4 GHz
- Pulse Repeat Frequency (PRF): 144 Hz
- Radar receiver antenna – targets table between distances: 115 cm.

Echo signals of the pulse radar targets which are small metal plaque, large metal plaque, large plexiglas plaque, corner reflector, sphere, cylinder were received to computer media by using audio card has 44 KHz sample frequencies.



**Fig. 1.** The algorithm of the intelligent system.

Pre-processing to obtain the feature vector was performed on the digitized, which were received to computer media by using audio card, in the following order:

(i) **Filtering:** The stored RTE signals were high-pass filtered to remove unwanted low-frequency components, because the RTE signals is generally in the range of 0.5-2 kHz. The filter is a digital FIR, which is a 50th-order filter with a cut-off frequency equal to 500 Hz and window type is the 51-point symmetric Hamming window.

(ii) **White de-noising:** White noise is a random signal that contains equal amounts of every possible frequency, i.e., its FFT has a flat spectrum [17]. The RTE signals were filtered from removing the white noise by using wavelet. The white de-noising procedure contains three steps [17]:

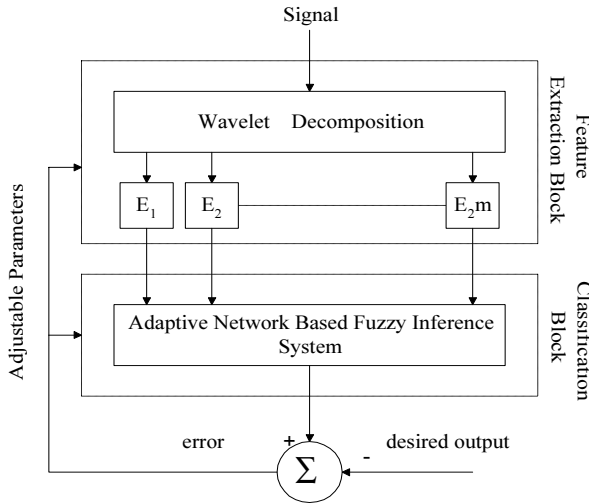
*1. Decomposition:* Computing the wavelet decomposition of the RTE signal at level 5 and using the Daubechies wavelet of order 4.

*2. Detail coefficient thresholding:* For each level from 1 to 5, soft thresholding is applied to the detail coefficients.

*3. Reconstruction:* Computing wavelet reconstruction based on the original approximation coefficients of level 5 and the modified detail coefficients of levels from 1 to 5.

### 3.2 Feature Extraction and Classification

Fig. 2 shows the WANFIS structure for classification of RTE signal waveform patterns from Radar experimental set. WANFIS embeds experiments about radar target



**Fig. 2.** The structure of wavelet adaptive network based fuzzy inference system for pattern classification.

classification topics of the expert human in intelligence system by using ANFIS structure. Feature extraction is the key for pattern recognition so that it is arguably the most important component of designing the intelligent system based on pattern recognition since the best classifier will perform poorly if the features are not chosen well. A feature extractor should reduce the pattern vector (i.e., the original waveform) to a lower dimension, which contains most of the useful information from the original vector.

The RTE waveform patterns from radar experimental set are rich in detail and highly non-stationary. After the data pre-processing has been realized, two steps are used to define the kind of these waveforms using MATLAB with the wavelet toolbox and the fuzzy toolbox:

1. *Wavelet Layer:* This layer is responsible for feature extraction from RTE waveform patterns from radar experimental set. The feature extraction process has two stages:

Stage 1 (Wavelet decomposition): For wavelet decomposition of the RTE waveforms, the tree structure was used  $m=5$  as level. In this study, eight targets that are small metal plaque, large metal plaque, large plexiglas plaque, corner reflector, sphere, the side part of cylinder, the lower part of cylinder, and the crosswise part of cylinder are used to obtaining the RTE signals. For each of these targets, three RTE signals that have different distances with radar transmitter antenna are used. Therefore, total numbers of the RTE signals, which are obtained from the radar experimental set, are 24. For wavelet decomposition of the RTE waveforms, the decomposition structure, reconstruction tree at level 5 as shown in Fig. 3 was used. Wavelet decomposition was applied to the RTE signal using the Daubechies-4 wavelet decomposition filters. Thus, obtaining two types of coefficients: one approximation coefficients  $cA$  and five –detail coefficients  $cD$ . A representative example of the wavelet decomposition of the radar echo signal of small metal plaque target was shown in Fig. 3.

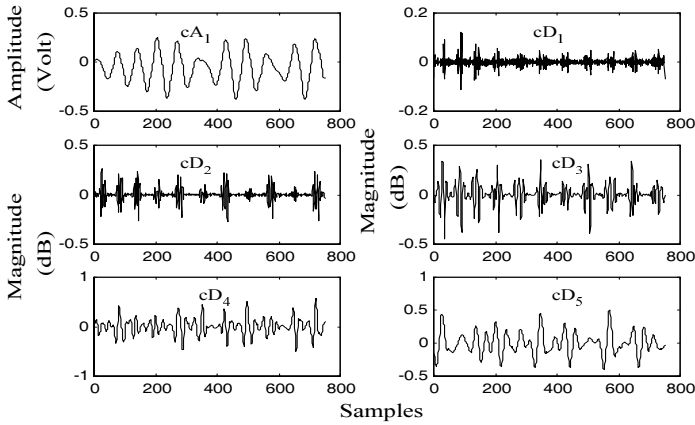


Fig. 3. The terminal node waveforms of wavelet decomposition at 5 levels of the RTE signal.

Stage 2 (Wavelet entropy): An Entropy-based criterion describes information-related properties for an accurate representation of a given signal. Entropy is a common concept in many fields, mainly in signal processing [6]. A method for measuring the entropy appears as an ideal tool for quantifying the ordering of non-stationary signals. We next calculated the norm entropy as defined in Eq. (3) of the waveforms at the terminal node signals obtained from wavelet decomposition

$$E(s) = \frac{\sum_i |s_i|^P}{N}, \tag{3}$$

where, the wavelet entropy  $E$  is a real number,  $s$  is the terminal node signal and  $(s_i)$  is the waveform of terminal node signals. In norm entropy,  $P$  is the power and must be such that  $1 \leq P < 2$ . During the WANFIS learning process, the  $P$  parameter is updated by using 0.1 increasing steps together with weights to minimize the error. The resultant entropy data were normalized with  $N=50$ . Thus, the feature vector was extracted by computing the 6-wavelet entropy values per RTE signal.

2. Adaptive Network Based Fuzzy Inference System (ANFIS) Layer: This layer realizes the intelligent classification using features from wavelet layer. The training parameters and the structure of the ANFIS used in this study are as shown in Table 1. These were selected for the best performance, after several different experiments, such as the number of input membership functions, the size of the ANFIS layers, value of the moment constant and learning rate, and type of the activation functions.

### 3.2.1 Structure of Wavelet Adaptive Network Based Fuzzy Inference System Developed in This Study

Both artificial neural network and fuzzy logic are used in ANFIS's architecture. ANFIS is consisted of if-then rules and couples of input-output, for ANFIS training is used learning algorithms of neural network [10-12].

For simplicity, we assume the fuzzy inference system under consideration has six inputs  $(x_1, x_2, x_3, x_4, x_5, x_6)$  that are wavelet entropies, which are given Eq. 3, of the obtained  $cA_1, cD_1, cD_2, cD_3, cD_4,$  and  $cD_5$  wavelet decomposition coefficients one

**Table 1.** ANFIS architecture and training parameters

The number of layers	5
	Input: 6, Rules number: 64, Output: 1
Type of Input Membership Functions	Bell-shaped
<i>Training parameters</i>	Hybrid Learning Algorithm (Back-propagation
Learning rule	for nonlinear parameters (a <sub>i</sub> , c <sub>i</sub> ) and Least square errors for linear parameters (p <sub>i</sub> , q <sub>i</sub> , r <sub>i</sub> , s <sub>i</sub> , ss <sub>i</sub> , pp <sub>i</sub> , u <sub>i</sub> )
Sum-squared error	0.0000001
Reaching epochs number to sum-squared error	472

output z. For a first order Sugeno fuzzy model, a typical rule set with base fuzzy if-then rules can be expressed as

$$\begin{aligned} &\text{If } x_1 A_1 \ x_2 B_1 \ x_3 C_1 \ x_4 D_1 \ x_5 E_1 \ x_6 F_1 \ \text{then} \\ &f_1 = px_1 + rx_2 + qx_3 + sx_4 + ssx_5 + ppx_6 + u \end{aligned} \tag{4}$$

Where, p, r, q, s, ss, pp, u are linear output parameters. The ANFIS’s architecture which has six inputs and one output. This architecture is formed by using five layer and sixty-four if-then rules:

**Layer-1:** Every node i in this layer is a square node with a node function.

$$\begin{aligned} O_{1,i} &= \mu_{A_i}(x), \text{ for } i=1,2, & O_{1,i} &= \mu_{B_{i-2}}(y), \text{ for } i=3,4 \\ O_{1,i} &= \mu_{C_{i-4}}(t), \text{ for } i=5,6 & O_{1,i} &= \mu_{D_{i-6}}(k), \text{ for } i=7,8 \\ O_{1,i} &= \mu_{E_{i-8}}(t), \text{ for } i=9,10 & O_{1,i} &= \mu_{F_{i-10}}(k), \text{ for } i=11,12 \end{aligned} \tag{5}$$

Where  $x_1, x_2, x_3, x_4, x_5, x_6$  are inputs to node i, and  $A_i, B_i, C_i, D_i, E_i, F_i$  are linguistic label associated with this node function. In order words,  $O_{1,i}$  is the membership function of  $A_i, B_i, C_i, D_i, E_i, F_i$ . Usually we choose  $\mu_{A_i}(x), \mu_{B_i}(y), \mu_{C_i}(t), \mu_{D_i}(k), \mu_{E_i}(t), \mu_{F_i}(k)$  to be bell-shaped with maximum equal to 1 and minimum equal to 0, such as

$$\mu_{A_i}(x), \mu_{B_{i-2}}(y), \mu_{C_{i-4}}(t), \mu_{D_{i-6}}(k), \mu_{E_{i-8}}(k), \mu_{F_{i-10}}(k) = \exp(-(|x_i - c_i|/a_i)^2) \tag{6}$$

Where  $a_i, c_i$  is the parameter set. These parameters in this layer are referred to as premise parameters.

**Layer-2:** Every note in this layer is a circle node labeled  $\Pi$  which multiplies the incoming signals and sends the product out. For instance,

$$O_{2,i} = w_i = \mu_{A_i}(x) \cdot \mu_{B_{i-2}}(y) \cdot \mu_{C_{i-4}}(t) \cdot \mu_{D_{i-6}}(k) \cdot \mu_{E_{i-8}}(k) \cdot \mu_{F_{i-10}}(k), i=1, 2, 3, \dots, 64 \tag{7}$$

Each node output represents the firing strength of a rule. (In fact, other T-norm operators that performs generalized AND can be used as the node function in this layer).

**Layer-3:** Every node in this layer is a circle node labelled N. The  $i^{th}$  node calculates the ratio of the  $i^{th}$  rules firing strength to the sum of all rule’s firing strengths:

$$O_{3,i} = \bar{w}_i = w_i / (w_1 + w_2 + \dots + w_{64}), i=1,2,3, \dots, 64 \tag{8}$$

**Layer-4:** Every node i in this layer is a square node with a node function

$$O_{4,i} = \bar{f}_i \cdot f_i = \bar{w}_i ( p_i x_1 + r_i x_2 + q_i x_3 + s_i x_4 + ss_i x_5 + pp_i x_6 + u_i ), i=1,2,3, \dots, 64 \tag{9}$$

Where,  $w_i$  is the output of layer 3, and  $\{p_i, q_i, r_i, s_i, ss_i, pp_i, u_i\}$  is the parameter set. Parameters in this layer will be referred to as consequent parameters.

**Layer-5:** The single node in this layer is a circle node labelled  $\Sigma$  that computes the overall output of WANFIS as the summation of all incoming signals:

$$O_{5,i} = \text{overall output} = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (10)$$

## 4 Experimental Results

We performed experiments using total 96 the RTE signals of small metal plaque, large metal plaque, large plexiglas plaque, corner reflector, sphere, the side part of cylinder, the lower part of cylinder, and the crosswise part of cylinder targets. For each of these targets, six numbers RTE signals were used that three of these signals have different distances with radar transmitter antenna and other three signals are noisy signals, which have different white-noise amplitudes (Signal / Noise Rate (SNR) = -2 dB, -3 dB, and -5 dB). 24 of these 96 signals were used for training and another part in testing the WANFIS. In these experiments, 100 % correct classification was obtained at the WANFIS training among the eight different target signal classes. It clearly indicates the effectiveness and the reliability of the proposed approach for extracting features from RTE signals. The WANFIS testing results are tabulated in Table 2.

**Table 2.** Performance of the intelligence system.

	Small Metal Plaque	Large Metal Plaque	Plexiglas Plaque	Corner Reflector	Sphere	The side part of cylinder	The lower part of cylinder	The crosswise part of cylinder
Total number of samples	9	9	9	9	9	9	9	8
Correct classification #	9	9	8	7	8	8	7	1
Incorrect classification #	-	-	1	2	1	1	2	-
The average recognition (%)	100	100	88.8	77.7	88.8	100	77.7	88.8

## 5 Discussions and Conclusion

In this study, we developed an intelligent system for the interpretation of the RTE signals using pattern recognition and the target recognition performance of this method demonstrated on the small metal plaque, large metal plaque, large plexiglas plaque, corner reflector, sphere, the side part of cylinder, the lower part of cylinder, and the crosswise part of cylinder targets. The tasks of feature extraction and classifi-

cation were performed using the WANFIS. The stated results show that the proposed method can make an effective interpretation. The performance of the intelligence system was given on Table 2.

The feature choice was motivated by a realization that WANFIS essentially is a representation of a signal at a variety of resolutions. In brief, the wavelet decomposition has been demonstrated to be an effective tool for extracting information from the RTE signals. The proposed feature extraction method is robust against to noise in the Rte signals.

In this paper, the application of the wavelet entropy in the wavelet layer of WANFIS to the adaptive feature extraction from RTE signals was shown. Wavelet entropy proved to be a very useful features for characterizing the RTE signal, furthermore the information obtained from the wavelet entropy is related to the energy and consequently with the amplitude of signal. This means that with this method, new information can be accessed with an approach different from the traditional analysis of amplitude of RTE signal.

The most important aspect of the intelligent system is the ability of self-organization of the WANFIS without requirements of programming and the immediate response of a trained net during real-time applications. These features make the intelligent system suitable for automatic classification in interpretation of the RTE signals. These results point out the ability of design of a new intelligence target recognition assistance system. The recognition performances of this study show the advantages of this system: it is rapid, easy to operate, and not expensive. This system offers advantage in military application. However, the position of the target and radar receiving antenna, which are used for data acquisition from the radar experimental set must be taken into consideration.

## References

1. Ahern, J., Delisle, G. Y., etc., Radar, Lab-Volt Ltd., vol. 1, p.p. 4-7 Canada, (1989).
2. Nelson, D. E., Starzyk, J. A. and Ensley, D. D. Iterated Wavelet Transformation and Signal Discrimination for HRR Radar Target Recognition IEEE Transaction on Systems, Man, And Cyberntics-Part A: Systems And Humans, Vol. 33, No. 1., (2002).
3. Mitchell, R. A., Hybrid Statistical Recognition Algorithm for Aircraft Identification, Univ. Dayton Press, Dayton, OH, (1997).
4. Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I., Image coding using wavelet transform, *IEEE Trans. Image Processing*, vol. 1, pp. 205–220, (1992).
5. Devaney, A. J., Raghavan, R., Lev-Ari, H., Manolakos, E., and Kokar, M., Automatic Target Detection and Recognition: A Wavelet Based Approach,” Northeastern Univ. Defense Technical Inform. Center, Tech. Rep. AD-A329 696, (1997).
6. Akay, M., Wavelet applications in medicine. *IEEE Spectrum*, 34, 50–56, (1997).
7. Strang, G. and T. Nguyen, T., *Wavelets and Filter Banks*. Wellesley, MA: Wellesley-Cambridge Press, (1996).
8. Devaney, A. J. and Hisconmez, B., Wavelet signal processing for radar target identification a scale sequential approach, in *Proc. SPIE Wavelet Applications*, vol. 2242, pp. 389–399, (1994).
9. Burrus, C. S., Gopinath, R. A. & Guo, H., Introduction to wavelet and wavelet transforms. NJ, USA: Prentice Hall, (1998).



10. Jang, J. S. R., Sun, C. T., Neuro-Fuzzy Modeling and Control, proceedings of the IEEE, vol. 83, No. 3, (1995).
11. Jang, J. S. R., ANFIS: Adaptive network-based fuzzy inference systems, IEEE, Trans. Syst., Man. and Cybern., vol. 23, pp. 665-685, (1993).
12. Avci, E., Turkoglu, I., "Modeling of Tunnel Diode by Adaptive-Network-Based Fuzzy Inference System", International Journal of Computational Intelligence, ISSN 1304-2386, Volume:1, Number:1, p.p. 231-233, (2003).
13. Zhang, Q., Benveniste, A., Wavelet Network, IEEE Trans. Neural Networks 3 (6), 889–898, (1992).
14. Thuillard, M., A Review of Wavelet Networks, Wavenets, Fuzzy Wavenets and Their Applications, ESIT'2000, 14 –15 September, Aachen, Germany, pp. 5–16, (2000).
15. Turkoglu, I., Arslan, A., Ilkay, E., An expert system for diagnosis of the heart valve diseases, Expert System with Applications 23 pp. 229-236, (2002).
16. Guler, I. and Ubeyli, E. D., Application of adaptive neuro-fuzzy inference system for detection of electrocardiographic changes in patients with partial epilepsy using feature extraction, Expert Systems with Applications, Volume 27, Issue 3, Pages 323-330, (2004).
17. Lou, X. and Loparo, K.A. K. A., Bearing fault diagnosis based on wavelet transform and fuzzy inference, Mechanical Systems and Signal Processing, Volume 18, Issue 5, Pages 1077-1095, (2004).
18. Ersahin, K., Yazgan, B., Satellite Image Fusion With Wavelet Decomposition, Eleco 2001 International Symposium Bursa, Turkey, (2001).

Part VIII

Robotics

# HMM-Based Gesture Recognition for Robot Control

Hye Sun Park<sup>1</sup>, Eun Yi Kim<sup>2</sup>, Sang Su Jang<sup>1</sup>, Se Hyun Park<sup>3</sup>,  
Min Ho Park<sup>4</sup>, and Hang Joon Kim<sup>1</sup>

<sup>1</sup> Department of Computer Engineering, Kyungpook National Univ., Korea  
{hspark, ssjang, hjkim}@ailab.knu.ac.kr

<sup>2</sup> Department of Internet and Multimedia Engineering, Konkuk Univ., NITRI\*, Korea  
eykim@konkuk.ac.kr

<sup>3</sup> School of Computer and Communication, Daegu Univ., Korea  
sehyun@daegu.ac.kr

<sup>4</sup> Information Technology Services, Kyungpook National Univ., Korea  
parkminho@msn.com

**Abstract.** In this paper, we present a gesture recognition system for an interaction between a human being and a robot. To recognize human gesture, we use a hidden Markov model (HMM) which takes a continuous stream as an input and can automatically segments and recognizes human gestures. The proposed system is composed of three modules: a pose extractor, a gesture recognizer, and a robot controller. The pose extractor replaces an input frame by a pose symbol. In this system, a pose represents the position of user's face and hands. Thereafter the gesture recognizer recognizes a gesture using a HMM, which performs both segmentation and recognition of the human gesture simultaneously [6]. Finally, the robot controller handles the robot as transforming the recognized gesture into robot commands. To assess the validity of the proposed system, we used the proposed recognition system as an interface to control robots, *RCB-1* robot. The experimental results verify the feasibility and validity of the proposed system.

## 1 Introduction

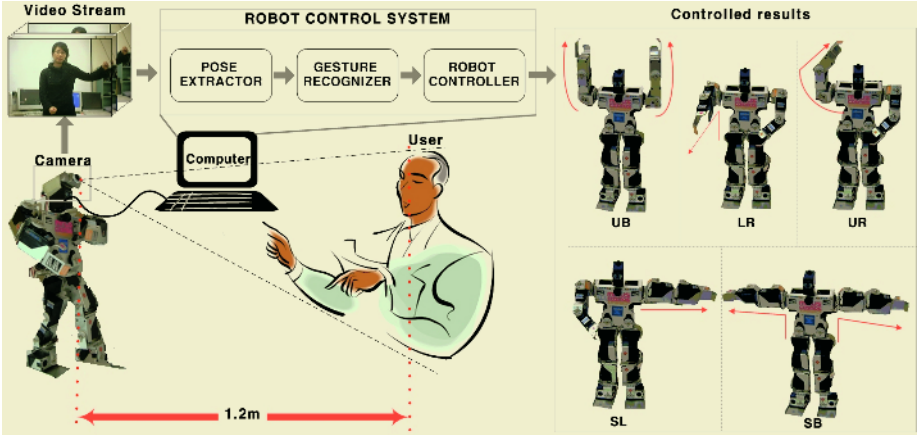
Recently, strong efforts have been carried out to develop intelligent and natural interfaces between users and computer systems based on human gestures [1-4]. Gestures provide an intuitive interface to both human and computer. Thus, such gesture-based interfaces can not only substitute the common interface devices, but also can be exploited to extend their functionality [5].

In this paper, we developed a gesture recognition system using hidden Markov model (HMM) for a robot control. Fig.1 shows the proposed system. The robot is always fixed at the desk with camera located in its head, and moves its arms according to the recognition results of human gestures. Our system consists of three modules: pose extractor, gesture recognizer, and robot controller. A pose extractor extracts a pose symbol for each frame. The pose is a 6-D vector, which consists of the positions of face, left and right hands. Thereafter the gesture recognizer receives a con-

---

\* Next-generation Innovative Technology Research Institute

tinuous pose symbol stream and recognizes a gesture. To recognition the gestures, we use the HMM developed in [6]. Generally HMMs have been widely used for many classification problems, as well as a gesture recognition problem, as HMMs have ability to model non-stationary signals or events. In this paper, the proposed HMM takes a continuous stream of pose symbols as an input and can automatically segments and recognizes human gesture. Finally, robot controller controls the robot as transforming the recognized gesture into robot commands.



**Fig. 1.** Outline of the proposed system: The robot is always fixed at the desk with camera located in its head, and moves its arms according to the gesture recognition results.

To assess the validity of the proposed system, we applied a real robot, *KHR-1* like Fig.1. The results show that the proposed system can provide a convenient and intuitive interface and it has a potential to apply for the robot control.

## 2 Definition of Gesture Commands

In our system, a robot is controlled by thirteen gestures. The gestures are that: {UP BOTH, UP LEFT, UP RIGHT, STRETCH BOTH, STRETCH LFET, STRETCH RIGHT, FOLD BOTH, LIFT BOTH, LIFT LEFT, LEFT RIGHT, DOWN BOTH, DOWN LEFT, DOWN RIGHT}. These thirteen gestures are related to the motions of robot arms. Fig.2 shows human gestures and the corresponding robot movements. Fig.2 (a) and (b) are FOLD BOTH command and UP BOTH command respectively. As shown in Fig. 2, the human acts on gesture like left side then the robot operates according to the gesture commands like right side.

## 3 Gesture Recognition Using a HMM

We use a HMM for gesture recognition. A continuous pose stream is used as an input of HMM, then the pose is defined as a vector to indicate the positions of face, left and right hands. We explain how a pose is extracted in subsection 3.1 and describe how the extracted pose stream would be recognized in subsection 3.2. Finally, we present how to control a robot using the recognized commands in subsection 3.3.

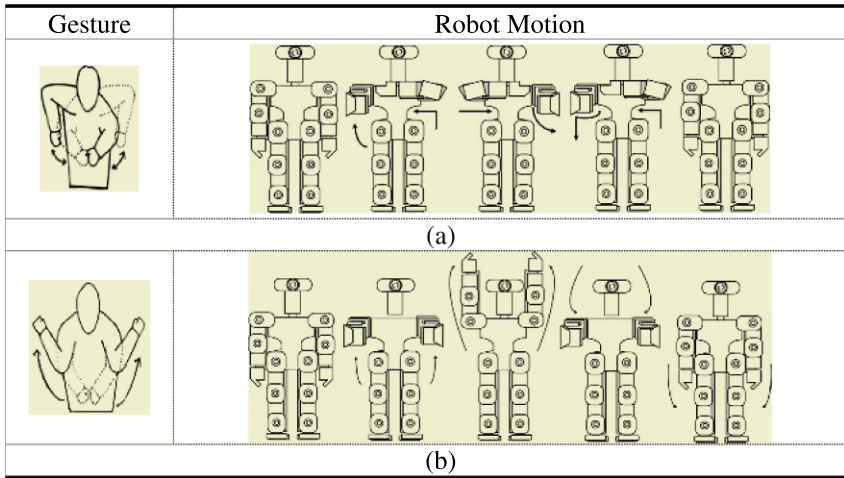


Fig. 2. Descriptions of human gestures and the corresponding robot motions.

### 3.1 Pose Extractor

A pose is to indicate the positions of face, left and right hands, thus it is represented as a vector  $P = (F_x, F_y, L_x, L_y, R_x, R_y)$ , where each element represents  $x$  and  $y$  coordinate of face, and left and right hands, respectively. The pose is extracted from each frame in the stream by three steps: skin-color region extraction, connected-component labeling, and template matching. This extraction process is illustrated in Fig.3.

An input frame is firstly divided into skin-color regions and non-skin-color regions using skin-color model that represented by 2-D Gaussian model [7].

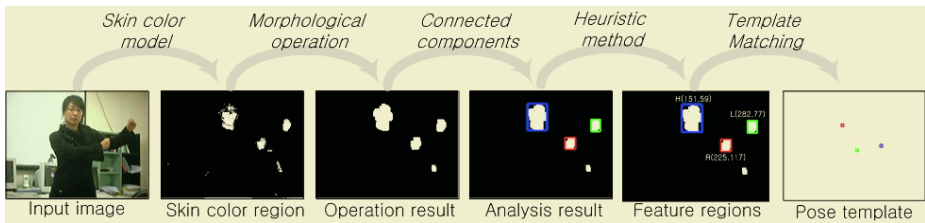


Fig. 3. The process of a pose extraction.

Thereafter, the results are filtered using connected-component labeling, and then positions of face, left hand, and right hand are obtained from 1<sup>st</sup> momentum of the respective components. Finally, the extracted position vector is classified into a pose symbol by a template matching: a position vector extracted from a frame is mapped to a pose symbol that has a smallest norm in the predefined symbol table.

Fig.4 shows the pose templates defined in this work. In this work, we assume that all gestures start and end with a same pose and that they can be distinguished by their distinctive pose. And the gesture has an intermediate pose between the start pose and its distinctive pose. Therefore a gesture is composed of four poses: {a starting pose,

an intermediate pose, a distinctive pose, and ending pose}. In Fig.4,  $T1$  is a template corresponding to a starting pose and ending pose as a ready pose. And the templates of from  $T2$  to  $T14$  are distinctive poses, and the others are intermediate poses.

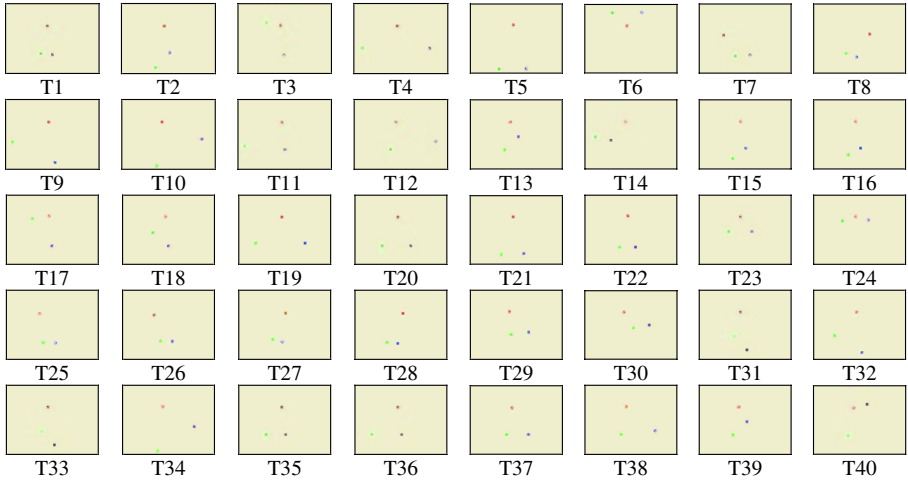


Fig. 4. Pose templates.

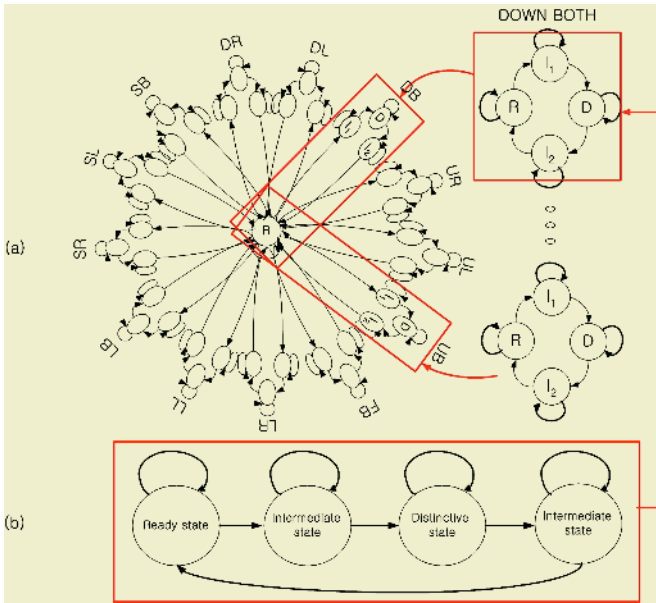
### 3.2 Gesture Recognizer

An HMM is used for gesture recognition, yet conventional HMMs usually work on isolated or pre-segmented sequences of input symbol sequences, and this type of segmentation is hard to implement.

Thus, to solve this segmentation problem and enhance the class discrimination a HMM architecture is developed that can automatically segment and recognize human gestures [6]. Fig.5 shows the proposed architecture which consists of a single HMM composed of thirteen gesture-specific HMMs that independently recognize certain gestures. In this work, we assumed that all gestures start and end with a same pose and that they can be distinguished by their distinctive pose. Therefore, we can combine all gestures in a single model, as each specific HMM shares a single ready state. In our model, each gesture state is composed of 4 states like Fig.5 (b). Then, the HMM recognizes a gesture from an input symbol stream using a 3<sup>rd</sup> state of each individual HMM as a path for the corresponding gesture. That is called *distinctive states*. And a gesture is detected and recognized, when the HMM passes this distinctive state.

The used HMM works as follows. First, the HMM starts with initial state probabilities  $s^0 = \{s_k^0\}$ , and continuously updates its state probabilities with the arrival of each input symbol as shown in Eq.1. If a value of any distinctive state in one of each gesture has higher state probability than predefined threshold, a gesture includes that state is detected and recognized.

$$s_n^t = \sum_{k=0}^{K-1} (s_k^{t-1} \times a_{kn}) \times b_{np} \quad , \quad s_n^t = \underline{s}_n^t / \sum_{k=0}^{K-1} \underline{s}_k^t \tag{1}$$



**Fig. 5.** An architecture of proposed HMM : (a) a single HMM, (b) a specific HMM.

In, Eq. 1,  $S=\{s_k\}$  denotes the state probability vector, where  $s_k$  is the state probability for the  $k^{th}$  state,  $a_{kn}$  is the probability of making a transition from state  $s_k$  to  $s_n$ ,  $b_{np}$  is the probability of emitting pose symbol  $v_n$  in state  $s_p$ .

Although the HMM's topology is somewhat complex, it is not too complicated to design, as we can design a set of small HMMs for each gesture independently, and combine those into a single HMM. And for using a single HMM for recognition system makes it easier to utilize the relations between gestures, we hope this topology to allow us more systematic approach to the co-articulation problem. Also, although the state probability is updated for every input symbol, we expect less computing load for this updating requires much less computation compared to the traditional pre-segmented matching process. And the recognition system responds to the input in real time, not waiting for all the isolated candidate sequence selection.

### 3.3 A Robot Controller

This module translates the recognized gestures into commands to control a robot. The controlled results are appeared as a motion of robot. The used robot, *KHR-1* can move his arms to various orientations using a servo control board, *RCB-1* equipped in a robot like Fig. 6(a). The *RCB-1* is operated by HeartToHeart1.0 which is shown Fig 6(b). To operate a robot, we control the HeartToHeart1.0 program. First, we make robot motions by motion generator in HeartToHeart1.0. Thereafter, we assign the motion number and memorize the assigned motion by motion controller. Given the recognized gesture, a robot controller controls robot motions by operating motion numbers.

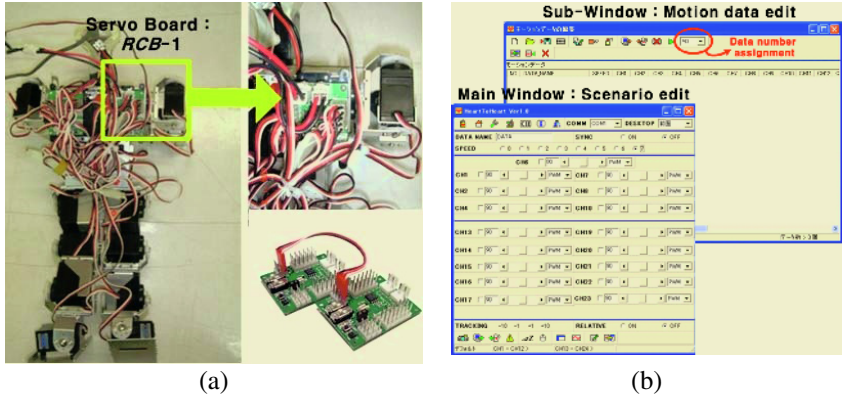


Fig. 6. Robot controller.

### 4 Experimental Results

In the experiments, a user is standing before a robot with a stationary background. The used robot is *KHR-1* which can controlled servo motor, *KRS-784*. The gestures of a user are captured by a digital video camera located in a robot head at an angle of fifteen degrees. The specification of our system is illustrated in Table 1.

Table 1. Specifications of the proposed robot control system.

Devices	Description	Devices	Description
Robot ( <i>KHR-1</i> )	Servo Motor : <i>KRS-784</i> , ICS 17 <sup>ch</sup>	Camera (MPC-C30 CCD)	350000 pixels
	Control Board : PIC16F873A		30 f/s, F=2.0mm
	Memory: 512K EEPROM		Interface: USB
	Voltage: DC 6V		f=4.9mm
	Size: 45x35(mm)		44 x 68 x 25mm
Weight: 12g	34g		
CPU		Pentium IV-2.0Ghz	
		OS: window XP	
		memory: 512M	

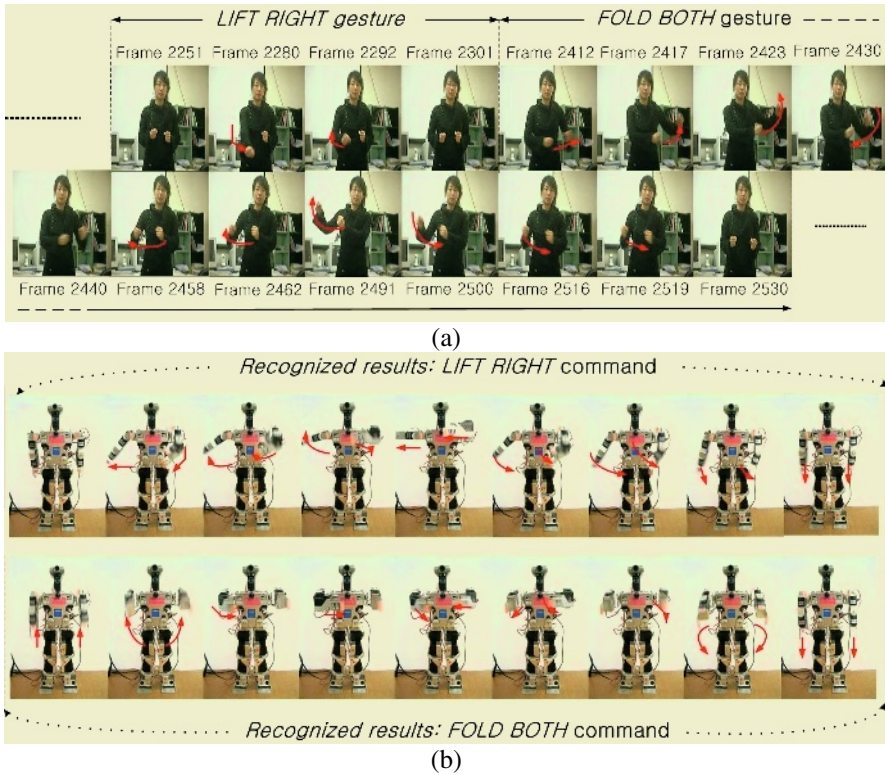
Using the proposed system, we control the robot’s arms. Then the examples of the control results are shown in Fig.7.

As shown in Fig.7, the proposed system segments and recognizes gestures from given input gesture frames, thereafter that recognized gesture commands operate a robot. Thus the robot moves according to recognized results.

Table 2 shows the recognition accuracy obtained for the proposed HMM approach. Overall, the approach classified 97.3% of the examples correctly and erred in 73 of the 2700 testing sequences. The proposed HMM’s most prominent error was a failure to recognize a ‘DL’, ‘DR’, ‘UR’ and ‘UL’ gesture, which was mistaken for a ‘LL’ or ‘LR’ gesture. Such errors occurred because the proposed HMM recognizes a gesture



from an input symbol stream using the third state of each individual HMM as a path for the corresponding gesture. But most errors happen from users who are unskilled for gesture commands.



**Fig. 7.** Examples of controlled robot by human gestures : (a) input frames (b) the controlled robot.

**Table 2.** Recognition results for the proposed HMM.

Gesture given	2700	Gestures recognized													No gesture
		LL	LB	LR	SR	SB	SL	FB	DB	DL	DR	UR	UL	UB	
LL	200	198	-	-	-	-	-	-	-	-	-	-	-	-	2
LB	200	-	199	-	-	-	-	-	-	-	-	-	-	-	1
LR	200	-	-	199	-	-	-	-	-	-	-	-	-	-	1
SR	200	-	-	-	197	-	-	-	-	-	-	-	-	-	3
SB	200	-	-	-	-	200	-	-	-	-	-	-	-	-	-
SL	200	-	-	-	-	-	198	-	-	-	-	-	-	-	2
FB	200	-	-	-	11	-	-	189	-	-	-	-	-	-	-
DB	200	-	-	-	-	-	-	-	192	-	-	-	-	-	8
DL	200	8	-	-	-	-	-	-	-	191	-	-	-	-	1
DR	200	-	-	7	-	-	-	-	-	-	192	-	-	-	1
UR	200	-	-	5	-	-	-	-	-	-	-	195	-	-	-
UL	200	5	-	-	-	-	-	-	-	-	-	-	195	-	-
UB	200	-	-	-	-	-	-	-	-	-	-	-	-	200	-
No gesture	100	3	-	2	-	-	-	5	2	3	3	-	-	-	82

## 5 Conclusions

In this paper, the gesture-based interface has been successfully implemented on the robot, *KHR-1*. To recognize a human gesture, we use a HMM which can automatically segments and recognizes the human gesture. Experimental result shows that the gesture-based interface provides a more convenient and intuitive to control a robot.

## Acknowledgement

This research was supported by the Daegu University Research Grant, 2004.

## References

1. Chao Hu, Max Qinghu Meng, Peter Xiaoping Liu, and Xiang Wang.: Visual Gesture Recognition for Human-Machine Interface of Robot Teleoperation, *IEEE/RSJ*, (2003) 1560~1565.
2. Milyn C. Moy.: Gesture-based Interaction with a Pet Robot, *AAAI*, (1999) 628 – 633.
3. Terence Fong and etc.: Novel interfaces for remote driving:gesture, haptic and PDA, *SPIE Telemanipulator and Telepresence Technologies VII*, (2000).
4. Oka, K., Sato, Y. and Koike, H.: Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems, *FGR*, (2002) 411-416.
5. Corradini, A.; Gross, H.-M.: Camera-based gesture recognition for robot control, *IJCNN* (2000).
6. H. S. Park, E. Y. Kim, H. J. Kim, “A Hidden Markov Model for Gesture Recognition,” *Pattern Recognition*, in review.
7. Jie Yang, Waibel, A.: A real-time face tracker, *WACV*, Vol. 15, no. 1. (1996) 142-147.

# PCA Positioning Sensor Characterization for Terrain Based Navigation of UVs

Paulo Oliveira\*

IST/DEEC and ISR, Torre Norte 8º andar  
Av. Rovisco Pais, 1, 1049 – 001 Lisbon, Portugal  
pjcro@isr.ist.utl.pt  
<http://www.isr.ist.utl.pt/~pjcro/indexi.html>

**Abstract.** Principal Component Analysis has been recently proposed as a nonlinear positioning sensor in the development of tools for Terrain Based Navigation of Underwater Vehicles [10]. In this work the error sources affecting the proposed unsupervised methodology will be enumerated, the stochastic characterization will be studied, and the attainable performance will be discussed. Based on a series of Monte Carlo experiments for a large set of synthesized terrains, conclusions will be drawn on the adequacy of the proposed nonlinear approach.

## 1 Introduction

Navigation systems design for long range missions of underwater vehicles (UVs) in unstructured environments, without resorting to external sensors, and with bounded error estimates, has been a major challenge in underwater robotics [6]. Unmodelled dynamics, time-varying phenomena, and the noise present in the sensor measurements continuously degrades the navigation system accuracy along time, precluding its use in a number of interesting applications. To overcome this limitation, external positioning systems have been proposed and successfully operated in the past, as extensively enumerated in [13], and integrated in navigation systems for underwater applications, as reported in the design examples found in [1, 14]. Unfortunately, all those positioning systems only locally provide accurate measurements (a few square kilometers), take long time to deploy, and are hard to calibrate, strongly constraining the area where the missions can take place, and ultimately the use of UVs.

One alternative central to this work has been exploited in the past: in the case where the missions take place in areas where detailed bathymetric data are available, the terrain information can aid to bound the error estimates of the navigation systems leading to Terrain Based, Terrain Reference, or Terrain Aided Navigation Systems. Applications with relative success have been reported in the past for air [2, 4], land [3] and underwater [5, 12] robotic platforms.

Extended Kalman Filtering has been the most common synthesis technique to tackle the terrain based navigation system design, as reported in [4, 12] and

---

\* Work supported by the Portuguese FCT POSI Programme under Framework QCA III and in the scope of project MAYA-Sub of the AdI.

the references therein. However, examples of performance degradation (including instability) on the proposed solutions have been reported by the same authors, precluding their use in general. Correlation techniques [2, 8], image matching techniques [12], and particle filters [5, 11] have also been proposed requiring high computational load, with limited performance and robustness.

This paper tries to elucidate on the adequacy of using unsupervised optimal processing techniques of random signals, namely Principal Component Analysis (PCA) (based on the Karhunen-Loève Transform) [7], to obtain a nonlinear positioning sensor instead of using a nonlinear estimator. The performance of the proposed sensor will be studied in a large set of terrains carefully chosen, providing bounds on the expected performance for the problem at hand.

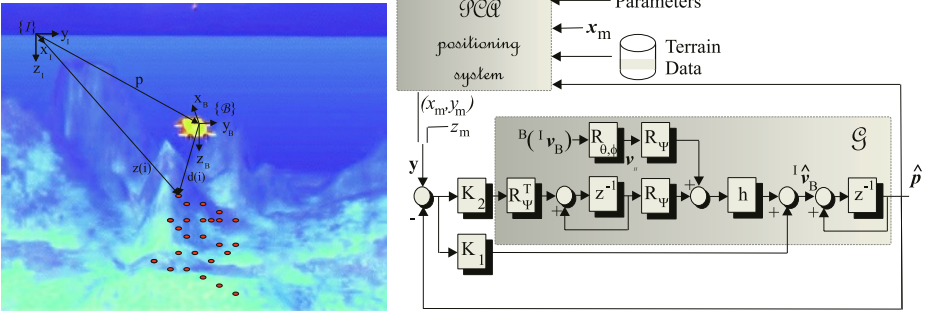
The paper is organized as follows: in section 2 the sensor package installed onboard is introduced and the underlying estimator structure is briefly described. Section 3 reviews the background on the Karhunen-Loève transform, basis for the principal component analysis of stochastic signals and details on the approach for the bathymetric data decomposition. In section 4 relevant terrains are discussed and the impact of PCA design parameters on the overall performance are enumerated and studied in detail. A stochastic characterization, resorting to a series of Monte Carlo experiments, is presented. Finally, in section 5 some conclusions are drawn on the adequacy of the proposed nonlinear approach and future work is unveiled.

## 2 UV Sensor Package and Navigation System

### 2.1 Notation and Sensor Package

Let  $\{\mathcal{I}\}$  be an inertial reference frame located at the pre-specified mission scenario origin with North, East, and Down axes (without loss of generality at mean sea level), as depicted in Fig. 1 and let  $\{\mathcal{B}\}$  denote a body-fixed frame that moves with an Underwater Vehicle (UV). The vehicle will be equipped with an Attitude and Heading Reference System (AHRS) that provides measurements of the attitude  $\lambda = [\phi \ \theta \ \psi]^T$ , i. e. the vector of roll, pitch, and yaw angles that parameterize locally the orientation of frame  $\{\mathcal{B}\}$  with respect to  $\{\mathcal{I}\}$ ,  ${}^{\mathcal{I}}\mathcal{R}(\lambda)$  and of the angular velocities expressed in body frame  ${}^{\mathcal{B}}({}^{\mathcal{I}}\omega_{\mathcal{B}})$ , i.e. body-fixed angular velocity. Note that since  $\mathcal{R}$  is a rotation matrix, it satisfies the orthogonality condition  $\mathcal{R}^T = \mathcal{R}^{-1}$  that is,  $\mathcal{R}^T\mathcal{R} = I$ . To complement the information available onboard the UV, a Doppler velocity log and a depth cell, providing measurements of  ${}^{\mathcal{B}}({}^{\mathcal{I}}\mathbf{v}_{\mathcal{B}})$  and the vertical coordinate  $z$ , respectively, are used.

To provide measurements for the PCA based positioning system a sonar ranging sensor is required, with a linear array of beams, where a bearing angle  $\epsilon$  associated with each of the beams is used. See Fig. 1 in detail, where the seafloor points sensed in several ranging measurements -  $d(i)$  - are depicted in red. Assuming, without loss of generality, that the sonar is installed at the origin of the reference frame  $\mathcal{B}$  pointing down, and the bearing angle lies in the transversal plane (containing the  $(y_B, z_B)$  axes), the  $i^{th}$  measurement can be geo-referenced in the inertial reference frame  $\mathcal{I}$  using



**Fig. 1.** Left: UV inertial and local coordinate frames. Mechanical scanning sonar range measurements. Right: Block diagram of the nonlinear estimator.

$$z(i) = \mathbf{p} + \frac{\mathcal{I}}{\mathcal{B}} \mathcal{R}(\lambda) \mathcal{R}_X(\epsilon) [0 \ 0 \ d(i)]^T, \quad (1)$$

where  $\mathcal{R}_X(\epsilon)$  is the rotation matrix, relative to the  $x_S$  axis, from the instantaneous sonar bearing to the UV reference frame  $\mathcal{B}$  and  $\mathbf{p} := [x \ y \ z]^T$  is the UV position relative to the inertial frame  $\mathcal{I}$ . It is important to remark that no support from other external systems/devices will be required.

## 2.2 Navigation System

An estimator for the state estimate  $\hat{\mathbf{z}}(k) = [\hat{\mathbf{p}}^T(k) \ \hat{\mathbf{b}}^T(k)]^T$ , corresponding to the vector obtained stacking the position estimate  $\hat{\mathbf{p}}$  and the bias estimates  $\hat{\mathbf{b}}$ , due to velocity sensor installation and calibration mismatches assumed constant or slowly varying, can be written resorting to the usual recursions for the Kalman filter:

$$\begin{cases} \hat{\mathbf{z}}^-(k+1) = \mathbf{A}(k)\hat{\mathbf{z}}^+(k) + \mathbf{B}_1(k)\mathbf{v}_H(k) \\ \mathbf{P}^-(k+1) = \mathbf{A}(k)\mathbf{P}^+(k)\mathbf{A}^T(k) + \mathbf{Q}(k), \end{cases} \quad (2)$$

where  $h$  is the sampling period,  $k$  describes in compact form the time instant  $t_k = kh$ , for  $k = 0, 1, \dots, T$  (the final mission time),  $\hat{\mathbf{z}}^-(k+1)$  is the predicted state variable estimate, and  $\mathbf{P}^-(k)$  is the covariance of the prediction estimation error, as detailed in [10]. Given a PCA position measurement, the state and error covariance updates,  $\hat{\mathbf{z}}^+(k)$  and  $\mathbf{P}^+(k)$ , respectively, will be given by

$$\begin{cases} \hat{\mathbf{z}}^+(k) = \hat{\mathbf{z}}^-(k) + \mathbf{K}(k)(y - \mathbf{C}(k)\hat{\mathbf{z}}^-(k)) \\ \mathbf{P}^+(k) = \mathbf{P}^-(k) - \mathbf{P}^-(k)\mathbf{C}^T(k)(\mathbf{C}(k)\mathbf{P}^-(k)\mathbf{C}^T(k) + \mathbf{R}(k))^{-1}\mathbf{C}^T(k)\mathbf{P}^-(k) \end{cases} \quad (3)$$

where  $\mathbf{K}(k) = \mathbf{P}^-(k)\mathbf{C}^T(k)(\mathbf{C}(k)\mathbf{P}^-(k)\mathbf{C}^T(k) + \mathbf{R}(k))^{-1} = [\mathbf{K}_p^T \ \mathbf{K}_b^T]^T$  is the Kalman filter gain, separable in two diagonal blocks and

$$\mathbf{R}(k) = \mathbf{f} r_{PCA}(k) \quad (4)$$

is the covariance of the observation noise. The factor  $\mathbf{f}$ , relating the PCA decomposition covariance and the sensor noise covariance is central for the problem at

end and will be the subject of a detailed stochastic characterization study. The resulting nonlinear estimator is represented in Fig. 1 on the right, with some abuse of notation.

### 3 Principal Component Analysis

Considering all linear transformations, the Karhunen-Loève (KL) transform allows for the optimal approximation to a stochastic signal, in the least squares sense. Furthermore, it is a well known signal expansion technique with uncorrelated coefficients for dimensionality reduction. These features make the KL transform interesting for many signal processing applications such as data compression, image and voice processing, data mining, exploratory data analysis, pattern recognition and time series prediction.

#### 3.1 PCA Background

Consider a set of  $M$  stochastic signals  $\mathbf{x}_i \in \mathcal{R}^N, i = 1, \dots, M$ , each represented as a column vector, with mean  $m_x = 1/M \sum_{i=1}^M \mathbf{x}_i$ . The purpose of the KL transform is to find an orthogonal basis to decompose a stochastic signal  $\mathbf{x}$ , from the same original space, to be computed as  $\mathbf{x} = \mathbf{U}\mathbf{v} + m_x$ , where the vector  $\mathbf{v} \in \mathcal{R}^N$  is the projection of  $\mathbf{x}$  in the basis, i.e.,  $\mathbf{v} = \mathbf{U}^T(\mathbf{x} - m_x)$ . The matrix  $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_N]$  should be composed by the  $N$  orthogonal column vectors of the basis, verifying the eigenvalue problem

$$\mathbf{R}_{xx}\mathbf{u}_j = \lambda_j\mathbf{u}_j, \quad j = 1, \dots, N, \quad (5)$$

where  $\mathbf{R}_{xx}$  is the ensemble covariance matrix, computed from the set of  $M$  experiments

$$\mathbf{R}_{xx} = \frac{1}{M-1} \sum_{i=1}^M (\mathbf{x}_i - m_x)(\mathbf{x}_i - m_x)^T. \quad (6)$$

Assuming that the eigenvalues are ordered, i.e.  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ , the choice of the first  $n \ll N$  principal components, leads to an approximation to the stochastic signals given by the ratio on the covariances associated with the components, i.e.  $\sum_n \lambda_n / \sum_N \lambda_N$ . In many applications, where stochastic multidimensional signals are the key to overcome the problem at hand, this approximation can constitute a large dimensional reduction and thus a computational complexity reduction. The advantages of PCA are threefold: i) it is an optimal (in terms of mean squared error) linear scheme for compressing a set of high dimensional vectors into a set of lower dimensional vectors; ii) the model parameters can be computed directly from the data (by diagonalizing the ensemble covariance); iii) given the model parameters, projection into and from the bases are computationally inexpensive operations of complexity  $\mathcal{O}(nN)$ .

#### 3.2 PCA Based Positioning System

Assume a mission scenario where bathymetric data are available and that a terrain based navigation system should be designed. The steps to implement a PCA based positioning sensor using this bathymetric data will be outlined next.

Prior to the mission, the bathymetric data of the area under consideration is partitioned in *mosaics* with fixed dimensions  $N_x$  by  $N_y$ . After reorganizing these two-dimensional data in vector form, e.g. stacking the columns, a set of  $M$  stochastic signals  $\mathbf{x}_i \in \mathcal{R}^N$ ,  $N = N_x N_y$ , results. The number of signals  $M$  to be considered depends on the mission scenario and on the *mosaic* overlapping. The KL transform is computed *a priori*, using (5) and (6), the eigenvalues are ordered, and the number  $n$  of the principal components to be used are selected, according with the required level of approximation. The following data should be recorded for latter use: *i*) the data ensemble mean  $m_x$ ; *ii*) the matrix transformation with  $n$  eigenvectors  $\mathbf{U}_n = [\mathbf{u}_1 \dots \mathbf{u}_n]$ ; *iii*) the projection on the selected basis of all the *mosaics*, computed using  $\mathbf{v}_i = \mathbf{U}_n^T(x_i - m_x)$ ,  $i = 1, \dots, M$ ; and *iv*) the coordinates of the center of the *mosaics*,  $(x_i, y_i)$ ,  $i = 1, \dots, M$ . During the mission, the last geo-referenced range measurements are packed and will constitute the input signal  $\mathbf{x}$  to the PCA positioning system. The following tasks should then be performed:

- i) compute the projection of the signal  $\mathbf{x}$  into the basis, using  $\mathbf{v} = \mathbf{U}_n^T(\mathbf{x} - m_x)$ ;
- ii) given an estimate on the actual horizontal coordinates of the UV position  $\hat{x}$  and  $\hat{y}$ , provided by the navigation system, search on a given neighborhood  $\delta$  the *mosaic* that verifies

$$\forall_i \|\hat{x} \hat{y}\|^T - [x_i \ y_i]^T\|_2 < \delta, \quad r_{PCA} = \min_i \|\mathbf{v} - \mathbf{v}_i\|_2; \quad (7)$$

- iii) given the *mosaic*  $i$  that is the closest to the present input, its center coordinates  $(x_i, y_i)$  will be selected as the position measurement.

Special attention should be taken next to two well known cases of poor performance: *i*) if the data correspond to white noise, the decomposition will result in equal eigenvalues, thus the use of  $n \ll N$  principal components will only explain the data covariance fraction  $n/N$ ; *ii*) in the case where data is spatially homogeneous (*flatland*) the decomposition is not unique, as any eigenvalue with null components associated, explains the (information empty) data.

## 4 PCA Positioning Sensor Performance

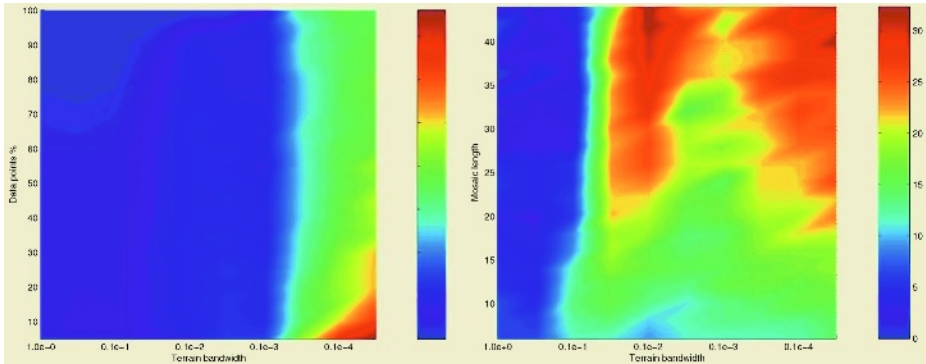
To study the performance of PCA as a positioning sensor a series of Monte Carlo tests were carried out using synthesized stochastic terrains

$$z(x, y) = \sum_{m=1}^{100} A(m) \sin(\Omega_x(m)x + \Phi_x(m)) \sin(\Omega_y(m)y + \Phi_y(m)),$$

where the spatial amplitude  $A(m) \approx \mathcal{N}(0, 1)$ , i.e. a white noise random variable with zero expected value and unitary variance, spatial frequency  $\Omega_i(m) = 2\pi f(m)$ ,  $i = \{x, y\}$ ,  $f(m) \approx \mathcal{U}(0, \bar{f})$ <sup>1</sup>, where  $\bar{f}$  is the maximum terrain bandwidth, and random spatial phase offsets  $\Phi_x(m)$  and  $\Phi_y(m)$ . The height is known

<sup>1</sup> Compact form to describe an uniformly distributed stochastic variable, in the interval expressed by the first and second arguments, respectively.

in a lattice  $x \in [\underline{x}, \bar{x}]$  and  $y \in [\underline{y}, \bar{y}]$ , at half meter intervals. Note that selecting  $\bar{f} = 1 \text{ m}$ , the Nyquist spatial frequency for the terrain representation in the present study, corresponds to white noise terrain and the *flatland* case is recovered in the limit where  $\bar{f} \rightarrow 0$ . Each experiment consists of randomly select a position  $(x, y)$  and bathymetric data (or some percentage of it) of the mosaic size according to (1), where each SONAR measurement is corrupted by zero mean white noise with  $\sigma_{SONAR} = 0.1 \text{ m}$ . The search in (7) is then performed over all the PCA data and the average and covariance of the position error are updated accordingly.



**Fig. 2.** Covariance relation  $\mathbf{f}$  from (4), for 1000 Monte Carlo experiments in each parameter combination, where  $M = 50 * 50$  mosaics were considered. Left: variations on the percentage of scanned points in the mosaic with dimensions  $N = 20 * 20$  versus terrain bandwidth. Right: variations on the mosaic length versus terrain bandwidth.

A number of parameters impact on the PCA positioning sensor performance. Next, some of the more relevant parameters are enumerated and the impact is discussed both based on Monte Carlo results and on the properties of the bathymetric terrain model:

**1. Number of Principal Components** - the increase on the number of components  $n$  increases the covariance accuracy explained (according to the ratio  $\sum_n \lambda_n / \sum_N \lambda_N$ ) [7]. Monte Carlo simulations revealed that a small number of components, are enough to explain in the excess of 95% of data covariance, thus validating the use of PCA as a low complexity positioning sensor.

**2. Number of Mosaics** - In the case where the neighborhood  $\delta \rightarrow \infty$  in (7), as considered in all this study, the performance of the overall PCA positioning system degrades linearly with the number of mosaics, due to the linear increase on the number of elements to be searched. In real applications, a careful choice of this parameter can improve the PCA performance, given an estimated position available from the estimator briefly introduced in section 2, as depicted in figure 1, on the right, thus bounding the positioning sensor error.

**3. Percentage of Points Scanned in a Mosaic** - Due to the velocity of propagation of sound in the water, only a fraction of the total *mosaic* area can be



scanned with a sonar. A graceful degradation on the performance is confirmed from the results, along any vertical line on the left of figure 2.

**4. Terrain Bandwidth** - This parameter is of utmost importance on the performance of the PCA positioning system, confirmed by the results depicted in Fig. 2. Note that on both left and right pictures, the white noise and the "flatland" cases were considered. The variation on  $f$  is nonlinear in both cases, thus prior to be used in real applications a multi model adaptive estimation strategy should be considered, specially when missions taking place on large areas are considered.

**5. Mosaic Dimension** - The mosaic dimension represents a compromise: small mosaic sizes increase the accuracy of the PCA positioning system, with an increase on the total number of mosaics. Large mosaic sizes diminish the accuracy and augments the correlation stored in each mosaic requiring an increase on the number of components. On the right part of Fig. 2, for a fixed number of components  $n$ , the performance degradation is evident, with a more severe increase in large mosaics. The impact on the performance is also nonlinear, thus providing an insight on strategies to tune the mosaic length selection.

The results obtained reinforce the usefulness of the proposed method as a basic positioning sensor, allowing the design of bounded accuracy underwater navigation and guidance systems. For a design example on the development of navigation tools, based on a PCA positioning sensor see [10].

## 5 Conclusions and Future Work

After performing a large number of Monte Carlo experiments with the proposed PCA positioning sensor, some conclusions can be drawn: the sensor is non-biased however it presents nonlinear characteristics for different terrains and PCA parameters'. In general, equal covariance in both dimensions were found, given the homogeneous definition of the set of terrains used. Thus, the results obtained pave the way to the use of the proposed sensor in real positioning applications for underwater robotics.

Future work will be carried out on the implementation of multi model adaptive estimator design and analysis tools for underwater navigation systems, where Doppler log/PCA and INS/PCA systems are of interest. It is important to remark that the design of navigation systems based on other geophysical sensors, such as magnetometers and gradiometers, is obvious.

## References

1. Alcocer, A., Oliveira, P., Pascoal, A.: "Study and Implementation of an EKF GIB-based Underwater Positioning System," IFAC Conference on Control Applications in Marine Systems CAMS04, Ancona, Italy, 2004.
2. Baker, W., Clem, R.: "Terrain Contour Matching (TERCOM) Primer," ASD-TR-77-61, Aeronautical Systems Division, Wright-Patterson AFB, Ohio, Aug. 1977 (AD B021328).

3. Crowley, J., Wallner, F., Sciele, B.: "Position Estimation Using Principal Components of Range Data," In Proceedings 1998 IEEE International Conference on Robotics and Automation, 1998.
4. Hostetler, L., Andreas, R.: "Nonlinear Kalman Filtering Techniques for Terrain-Aided Navigation," IEEE Transactions on Automatic Control, vol. AC-28, No. 3, March 1983.
5. Karlsson, R., Gustafsson, F.: "Particle Filter for Underwater Terrain Navigation," 2003 IEEE Workshop on Statistical Signal Processing, September 2003.
6. Leonard, J., Bennett, A., Smith, C., Feder H.: "Autonomous Underwater Vehicle Navigation," MIT Marine Robotics Laboratory Technical Memorandum, USA, 1998.
7. Mertins A.: Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications, John Wiley & Sons, 1999.
8. Nygren, I., Jansson, M.: "Robust Terrain Navigation with the Correlation Method for High Position Accuracy," in OCEANS 2003 Marine Technology and Ocean Science Conference San Diego, CA, Vol. 3, September 2003.
9. Oliveira, P.: Periodic and Nonlinear Estimators with Applications to the Navigation of Ocean Vehicles, Ph.D. Thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, July 2002.
10. Oliveira, P.: "Terrain Based Navigation Tools for Underwater Vehicles using Eigen Analysis," accepted for publication in the 16th IFAC World Conference, July 2005.
11. Pham, D., Dahia, K., Musso, C.: "A Kalman-Particle Kernel Filter and its Application to Terrain Navigation," Proceedings of the Sixth International Conference of Information Fusion, vol. 2 , July 2003.
12. Sistiaga, M., Opderbecke, J., Aldon, M., Rigaud, V.: "Map Based Underwater Navigation using a Multibeam Echosounder," OCEANS '98 Conference Proceedings, vol. 2, September 1998.
13. Vickery, K.: Acoustic Positioning Systems - A Practical Overview of Current Systems, Proceedings Of The 1998 Workshop on Autonomous Underwater Vehicles-AUV'98, August 1998.
14. Whitcomb, L., Yoerger, D., Singh, H.: "Combined Doppler/LBL Based Navigation of Underwater Vehicles," 11th International Symposium on Unmanned Untethered Submersible Technology (UUST99), Durhan, USA, 1999.

# Monte Carlo Localization Using SIFT Features

Arturo Gil, Óscar Reinoso, Asunción Vicente,  
César Fernández, and Luis Payá

Área de Ingeniería de Sistemas y Automática  
Universidad Miguel Hernández  
Avda. del Ferrocarril s/n  
03202 Elche (Alicante), Spain  
[arturo.gil@umh.es](mailto:arturo.gil@umh.es)  
<http://lorca.umh.es/isa/es>

**Abstract.** The ability of finding its situation in a given environment is crucial for an autonomous agent. While navigating through a space, a mobile robot must be capable of finding its location in a map of the environment (i.e. its pose  $\langle x, y, \theta \rangle$ ), otherwise, the robot will not be able to complete its task. This problem becomes specially challenging if the robot does not possess any external measure of its global position. Typically, dead-reckoning systems do fail in the estimation of robot's pose when working for long periods of time. In this paper we present a localization method based on the Monte Carlo algorithm. During the last decade this method has been extensively tested in the field of mobile Robotics, proving to be both robust and efficient. On the other hand, our approach takes advantage from the use of a vision sensor. In particular, we have chosen to use SIFT features as visual landmarks finding them suitable for the global localization of a mobile robot. We have successfully tested our approach in a B21r mobile robot, achieving to globally localize the robot in few iterations. The technique is suitable for office-like environments and behaves correctly in the presence of people and moving objects.

## 1 Introduction

The skill of navigating through an environment is a key aspect for a mobile robot. A mobile robot must be capable of travelling from a starting point in space, say  $A$  to a final point  $B$ . Frequently, the space traversed by the robot will be unstructured, changing and with people moving around. First, the mobile agent must plan a trajectory that starts at point  $A$  and ends at point  $B$ . The set of algorithms that solve this problem are often referred as path planning techniques. These techniques use a map of the environment to find the best path that reaches a particular destination from a given start location. While the robot moves along the planned path it needs to know its position/orientation, otherwise the mobile agent would not be able to follow it. Naively, the position/orientation of the robot can be determined using dead-reckoning, that is, using odometry sensors. However, these sensors lack of accuracy when used for long periods of time, due

to wheel slippage, drifts and other problems. GPS and inertial systems offer an alternative to dead reckoning, but have a drawback: Often mobile robots working in indoor environments cannot receive the GPS signal properly. Localization techniques are used instead, in order to find the position of the mobile agent in space. Most of them try to match salient characteristics of the space sensed by the robot with the same characteristics in the map, thus localizing the robot. For example, in [1] a CCD camera and a laser rangefinder are used to find vertical and horizontal structures in the space that surrounds the robot (i.e. corners and walls). Those structures are then matched against a map of the environment. As a result, the robot can localize itself in the map.

Our approach to localization is based on the Monte Carlo algorithm. During the last decade this method has been extensively tested in the field of mobile Robotics, proving to be both robust and efficient. On the other hand, our work is based on a stereo vision system, which allows us to compute the relative distance of the robot to significant points of the space. These significant points of space are usually called landmarks, and are the basis that enables the robot to deduce its location in a previously built map of the environment. In particular, we have chosen to use SIFT features as visual landmarks finding them suitable for the global localization of a mobile robot.

In section 2 we relate our work with previous implementations. Section 3 explains the use of SIFT features, which have been used previously in robotics applications, such as [2] and [3]. Next, in section 4 we will explain the basics of Monte Carlo localization. The integration of SIFT features in Monte Carlo localization will be also explained in section 4. Section 5 describes the experimental setup used to test the MCL algorithm together with the use of visual SIFT features. Finally, in section 6 we analyze the main results that we have obtained and propose future work related to our investigation.

## 2 Relation to Prior Work

As stated previously, a mobile robot must not rely uniquely on its odometry information to estimate its position: It must use the information provided by its sensors to observe the space that surrounds it and relate those observations with a map of the environment.

Vision sensors have been used by different groups for localization tasks. For example, in [4] Neira et al. use the information provided by a CCD camera and a laser rangefinder and then extract the significative characteristics from the space surrounding the robot. Using an EKF, those characteristics are matched with a previously built map of the environment, thus permitting the estimation of the robot's location. Olson [5], proposes the use of salient points in stereo images extracted using the Förstner interest operator. Afterwards, the 3D position of each point is calculated an ego-motion measure is estimated by matching the points accross successive images. This approach reduces significantly the error in tracking robot's position, however, it does not provide a solution for the global localization problem. In [2] and [3] stereo vision is used to track 3D visual landmarks extracted from an unstructured environment, in particular, SIFT features

are used as visual landmarks. During an exploration phase, the robot extracts the SIFT features from stereo images (actually a trinocular stereo system), calculating their 3D position in space, and stores them in a database (which conforms the map). Later, when the robot is navigating, the stored SIFT features are found in the environment and the relative distance to them is computed. Finally, its pose is calculated by means of a Hough transform.

In our work, we take an approach similar to that of [2] and [3]. That is, we extract SIFT landmarks from the environment and store them in a database. SIFT landmarks are characterized using a descriptor. This means that the same landmark can be recognised by the robot when it navigates through the environment, hence, this enables the use of SIFT features for the global localization problem. However, our localization algorithm differs greatly from the one cited previously. Indeed, we use a particle filter approach inspired in the work exposed in [6], [7] and [8], which has proved to be both fast and robust.

### 3 Sift Features

SIFT (Scale Invariant Feature Transform) features were developed by Lowe for image feature generation, and used initially in object recognition applications (See [9] and [10] for details). Lately, SIFT features have been used in robotic applications ([2], [3]), showing its suitability for localization and SLAM tasks. The features are invariant to image translation, scaling, rotation and partially invariant to illumination changes and affine projection. Thus, this enables the same point in space to be viewed from different poses of the robot, which may occur while the robot moves around its workplace, thus providing information for the localization process.

SIFT features are located at maxima and minima of a difference of Gaussian function applied in scale space. They can be computed by building an image pyramid with resampling between each level. SIFT locations are extracted by means of successive filtering. The input image is first convolved with a Gaussian function of  $\sigma = \sqrt{2}$ , resulting in image  $A$ . Next, the image is further convolved with a Gaussian function, yielding image  $B$ . SIFT locations are extracted as maxima and minima from the image  $C = A - B$ .

The SIFT locations extracted by this procedure can be understood as significant points in space that are highly distinctive. The next step needed is to describe that point in space, so that the robot can be capable of recognising it in a later stage, while it navigates through the environment. One simple solution would be to sample the image around the key location and store the values in a matrix. Then, a correlation measure could be used in order to identify the feature. However, this descriptor is very sensitive to illumination and 3D viewpoint changes, hence this solution does not produce valid results. In our application, we used a descriptor similar to the one proposed in [10], based on local image gradients, which behaves correctly with illumination and viewpoint changes. Once the SIFT location is calculated, we assign an orientation to each feature, based on local image properties. By doing this we can represent the descriptor relative to this orientation, thus achieving variance to image rotation.

## 4 Monte Carlo SIFT Localization

In robot localization we are interested in estimating the pose of the vehicle (typically, the state  $\mathbf{x} = \langle x, y, \theta \rangle$ ) using a set of measurements  $Z^k = \{\mathbf{z}_k, i = 1 \dots k\}$  from the environment and a set of actions  $\mathbf{u}_k$  performed. This can be stated in a probabilistic way, that is: Localization aims at estimating a belief function  $p(\mathbf{x})$  over the space of all possible poses, conditioned on all data available until time  $k$ , that is:  $p(\mathbf{x}_k | Z^k)$ . The estimation process is usually done in two phases:

**Prediction Phase:** In this phase, a motion model is used to calculate the probability density function (PDF)  $p(\mathbf{x}_k | Z^{k-1})$ , taking only motion into account. Usually it is assumed that the current state  $\mathbf{x}_k$  is only dependent on the previous state  $\mathbf{x}_{k-1}$  and a control input  $\mathbf{u}_{k-1}$ . The motion model is specified in the form of the conditional density:  $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ . The prediction is then obtained by integration:

$$p(\mathbf{x}_k | Z^{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1} | Z^{k-1}) d\mathbf{x}_{k-1} \quad (1)$$

**Update Phase:** In the second phase, a measurement model is used to incorporate information from the sensors and obtain the posterior PDF  $p(\mathbf{x}_k | Z^k)$ . The measurement model is given in terms of a probability  $p(\mathbf{z}_k | \mathbf{x}_k)$  which provides the likelihood of the state  $\mathbf{x}_k$  supposing that a particular measurement  $\mathbf{z}_k$  was observed. The posterior density  $p(\mathbf{x}_k | Z^k)$  can be calculated using Bayes' Theorem as follows:

$$p(\mathbf{x}_k | Z^k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | Z^{k-1})}{p(\mathbf{z}_k | Z^{k-1})} \quad (2)$$

The process is repeated recursively after update phase. The knowledge about the initial state at time  $t_0$  is represented by  $p(\mathbf{x}_0)$ . In the case of global localization, where the pose of the vehicle is totally unknown,  $p(\mathbf{x}_0)$  is represented by a constant function over the space of all possible poses.

Note that in expressions 1 and 2 nothing is said about the representation of the PDF. This fact leads to a series of different algorithms that are based on the above prediction-update scheme, mainly: The Kalman filter, Markov grid-based localization and Monte Carlo localization. The Kalman filter does not solve for the global localization. On the other hand, Markov grid-based localization requires large amounts of memory and computation time. Hence, our approach relies on the Monte Carlo localization method.

### 4.1 Monte Carlo Localization

Monte Carlo localization can be included in a set of algorithms called particle filters, which have had a great development during last decade (e.g. [7], [6] and [11]). In Monte Carlo localization (MCL for short), the PDF  $p(\mathbf{x})$  is represented

by a set of  $M$  random samples  $\chi_k = \{x_k^i, i = 1 \dots M\}$  extracted from it. Each particle can be understood as a hypothesis of the true state of the robot (i.e. its pose  $\langle x, y, \theta \rangle$ ). The algorithm is calculated in a Prediction-update fashion, as stated before.

**Prediction Phase:** A set of particles  $\chi_k$  is generated based on the set of particles  $\chi_{k-1}$  and a control signal  $u_k$ . This step uses the motion model  $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$  and applies it to every particle in set  $\chi_k$ . As a result, a new set of particles  $\chi'_k$  is generated, which represents the density  $p(\mathbf{x}_k | Z^{k-1})$ .

**Update Phase:** In this second phase, we take into account an observation  $z_k$  made by the robot. For each particle in the set, a weight  $\omega_k^i$  is computed (frequently called Importance Factor). This weight is calculated using the observation model  $\omega_k^i = p(z_k | x_k^i)$  resulting in the set  $\bar{\chi}_k = \{x_k^i, \omega_k^i\}$ . Finally the resulting set  $\chi_k$  is calculated by resampling with replacement from the set  $\bar{\chi}_k$ , where the probability of resampling each particle is given by its importance weight  $\omega_k^i$ . Finally, the set  $\chi_k$  represents the distribution  $p(\mathbf{x}_k | \mathbf{Z}_k)$ .

The prediction-update phases are repeated recursively. To localize the vehicle globally, the initial set of particles is spreaded randomly over the entire state space. See [11], [6] and [7] for details.

## 5 Experimental Results

In this section we report the Monte Carlo localization technique that has been tested together with SIFT features in an office-like environment. A B21r robot equipped with a calibrated stereo head was used for the experiments. In Fig. 1, an image of the environment is shown. The environment is characterized for being frequently traversed by people.

The experiment can be divided in two phases: A) Environment exploration and map creation, and B) Localization.

### 5.1 Environment Exploration and Map Creation

In this first phase, the robot was commanded to move along the environment, varying its position and orientation. Simultaneously, images were captured with both cameras and processed to extract SIFT features. Next, features extracted in the left image were matched with the ones found in right image. The following restrictions were applied during this process:

- Epipolarity restriction: The feature location in the right image must be placed in the same row as the in the left image. In practice, this condition was relaxed, permitting a maximum  $\pm 2$  pixel displacement.
- SIFT restriction: The euclidean distance between two SIFT descriptors must not surpass a certain threshold (determined experimentally).

Each time a SIFT feature is matched correctly in both images, its position relative to the robot is calculated using stereo vision. In addition, the position



**Fig. 1.** B21r robot during data acquisition in the environment.

of the SIFT feature in space is determined relative to a global frame. In order to minimize the error in robot's pose, the exploration phase was performed in several runs. Besides, in order to minimize the error in robot's odometry, a tracking procedure of the landmarks was used (similar to the exposed in [2]). The information gathered is stored in a database, which constitutes the map.

## 5.2 Localization

During localization, the robot navigates along the environment. Meanwhile, the robot captures images with its cameras, processes them and finds SIFT features. Next, a matching procedure is taken, in order to find the relative position of the feature. Afterwards, the robot tries to match any feature found with its correspondence in the database. The euclidean distance between SIFT descriptors is used in order to find corresponding features. Some issues appear during this process:

- The correspondence between the observed feature and a stored landmark is not direct. The robot may confuse one landmark for another, thus providing erroneous information for the localization algorithm.
- The use of visual landmarks allows us to localize the robot in populated environments. It is not necessary for the robot to visualize all SIFT points in the scene. The robot may only detect a few points and still will be capable of finding its pose in the map.

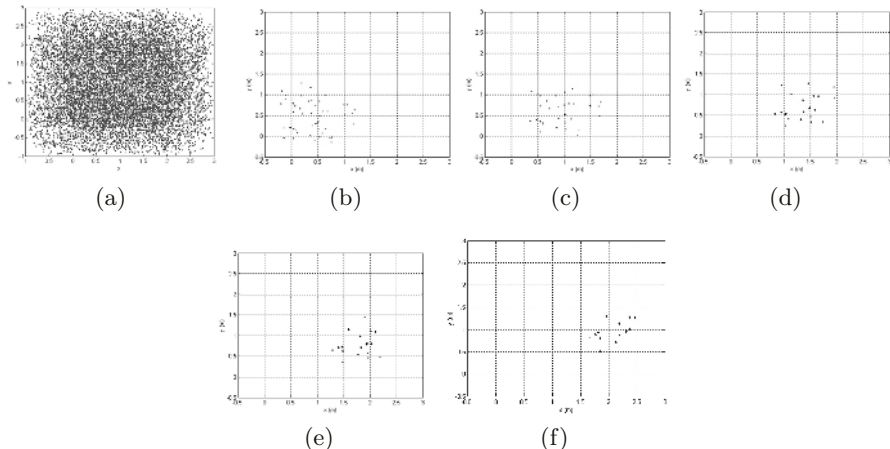
The parameters in the observation model were adjusted by experience. It is worth mentioning that the parameters in motion and observation models have a great influence in the convergence properties of MCL algorithm. A general trend is to inflate both motion and observation, which speeds up the localization process and avoids the possibility of losing track of the robot ([6], [11]).

In Fig. 2 a global localization process is shown. First, in Fig. 2(a) a random set of particles is spreaded over the entire space state (we show only the  $\langle x, y \rangle$  components for clarity). In the following figures a series of prediction-update phases are shown. Finally, in Fig. 2(f) the particles gather around the last robot position, hence localizing it. In Table 1 we show a comparison between odometry



**Table 1.** Odometry vs MCL comparison.

	Odometry			MCL		
	$x_{rob}$ (m)	$y_{rob}$ (m)	$\theta_{rob}$ (rad)	$x_{rob}$ (m)	$y_{rob}$ (m)	$\theta_{rob}$ (rad)
Fig. 2(b)	0.307	0.342	0.403	0.275	0.367	0.397
Fig. 2(c)	0.802	0.541	0.403	0.974	0.608	0.391
Fig. 2(d)	1.268	0.732	0.403	1.347	0.802	0.394
Fig. 2(e)	1.714	0.885	0.403	1.753	0.923	0.394
Fig. 2(f)	2.072	1.065	0.403	2.093	1.071	0.395



**Fig. 2.** MCL localization progress. Image a) shows the set of particles spreaded around the entire state space. From image and b) to f) the localization progress is presented. Finally, the robot is localized in image f).

and the MCL estimation. Odometry can be used as a good measure of the robot’s pose when used in short displacements, thus permits us to compare it with the result of the MCL estimation.

The algorithm was tested during several runs through the environment. We obtained similar results to the ones showed, achieving to localize the robot quite accurately in a few steps.

## 6 Discussion and Future Work

This paper describes a localization method based on the Monte Carlo algorithm in combination with visual landmarks. In particular, SIFT features have been used as visual landmarks, finding them suitable for the global localization problem. Our approach has been implemented on a mobile platform and tested in a

real environment. Good results have been achieved, proving the effectiveness of our solution. However, we plan to further test the algorithm for longer periods of time. During our experiments, we found that some SIFT features found in the environment lacked of stability: They were found from a robot's pose, but could not be detected from elsewhere. To solve this, we plan to track features for consecutive frames, hence ensuring that the feature found is stable and can be detected from different viewpoints.

Note that, in case a sudden change in robot's position may occur (i.e. the robot hits an obstacle), the algorithm could fail in localizing the robot. If the particles concentrate around a certain position and the observation model suggests that the robot is elsewhere, then the weight  $\omega_k^i$  may be zero for every particle in the set. This can be avoided by the injection of a random set of particles in each iteration of the MCL algorithm. We plan to add this feature in a future approach. On the other hand, as a future work we plan to apply a version of the MCL algorithm for multi-robot localization. We consider that the localization problem can be solved efficiently using several robots in combination with the method exposed in this paper.

## References

1. J. Neira J. D. Tardós J. Castellanos, J. M. Martínez. Experiments in multisensor mobile robot localization and map building. *3rd IFAC Symposium on Intelligent Autonomous Vehicles*, 1998.
2. J. Little S. Se, D. Lowe. Vision-based mobile robot localization and mapping using scale-invariant features. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2051–2058, 2001.
3. J. Little S. Se, D. Lowe. Global localization using distinctive visual features. *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems EPFL*, 2002.
4. J. Horn G. Schmidt J. Neira, J. D. Tardós. Fusing range and intensity images for mobile robot localization. *IEEE Transactions on Robotics and Automation*, 15(1):76–83, 1999.
5. M. Schoppers M. W. Maimone C. F. Olson, L. H. Matthies. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, (43):215–229, 2003.
6. F. Dellaert S. Thrun D. Fox, W. Burgard. Monte carlo localization: Efficient position estimation for mobile robots. *Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 2000.
7. W. Burgard S. Thrun F. Dellaert, D. Fox. Monte carlo localization for mobile robots. *IEEE International Conference on Robotics and Automation (ICRA99)*, 1999.
8. J. Guivant H. Durrant-Whyte E. Nebot, F. Masson. Robust simultaneous localization and mapping for very large outdoor environments. *Proceedings of the 8th International symposium on Experimental Robotics (ISER '02)*, 2002.
9. D. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, 2, 1999.
10. D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110, 2004.
11. W. Burgard F. Dellaert S. Thrun, D. Fox. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

# A New Method for the Estimation of the Image Jacobian for the Control of an Uncalibrated Joint System

Jose M. Sebastián<sup>1</sup>, Lizardo Pari<sup>1</sup>, Carolina González<sup>1</sup>, and Luis Ángel<sup>2</sup>

<sup>1</sup>Departamento de Automática (DISAM), Universidad Politécnica de Madrid  
C/ José Gutiérrez Abascal, 2, 28006 Madrid, Spain  
{jsebas, lpari, cgpascual}@etsii.upm.es

<sup>2</sup>Escuela de Ingeniería Electrónica, Universidad Pontificia Bolivariana  
Km 7, Via Piedecuesta, Bucaramanga, Colombia  
langel@etsii.upm.es

**Abstract.** This paper describes various innovative algorithms for the on-line estimation of the image jacobian, a matrix which linearly relates joint velocity and image feature velocity. We have applied them successfully to the static visual control of an uncalibrated 3 DOF joint system, using two weakly calibrated fixed cameras. The proposed algorithms prove to be particularly robust when image features are calculated with an average level of noise, and our results are clearly better than those obtained for already existing algorithms in specialized literature.

## 1 Introduction

The problem of developing tasks in robotic systems under structured environments, with the presence of objects whose position and orientation is perfectly known, has been extensively studied. However, operation in unknown and dynamic environments involves a large number of additional difficulties not completely solved at present. Vision systems can provide extremely useful information in these environments, since they offer information about which objects are present in the taskspace and, more importantly, their position, orientation and velocity can be determined precisely enough.

The use of vision sensors to close the control loop of a robot is known as Visual Servoing. Some of the most comprehensive surveys are those described in [1], [2], [3]. Visual servoing systems are mainly classified attending to their control scheme. Thus there is Position Based Visual Servoing (PBVS), where the error signal and control law are specified in Cartesian coordinates according to the desired and current position and orientation. This is also known as 3D visual servoing. On the other hand, in Image Based Visual Servoing (IBVS) the error signal and control law are specified in the image space according to the desired and current visual features. This is known as 2D visual servoing. This method implies calculating or estimating the Image Jacobian, which linearly relates image feature velocity and robot joint velocity.

In this paper we introduce a method for the on-line estimation of the image jacobian with two fixed cameras overlooking the scene, without need for Euclidean calibration of the cameras nor kinematic calibration of the robot. Its main novelty consists

in the use of the fundamental matrix in the calculation of the image jacobian, which allows a more robust estimation in the detection of image features in the presence of noise. Firstly we detail the terminology and theoretical concepts used in the paper, then we put forward the innovative algorithms proposed, and finally we describe our experiments and conclusions.

## 2 Image Jacobian

Assume that a robot or positioning system is observed from one or various fixed views. Let  $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_p]^T$  be the  $p$ -dimensional vector that represents the position of the end effector in a Cartesian coordinate system. Let  $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T$  be the  $n$ -dimensional vector that represents the joint position of the robot. Let  $\mathbf{s} = [s_1 \ s_2 \ \dots \ s_m]^T$  be the  $m$ -dimensional vector that represents the image features (for example the coordinates of a point in one or both images).

The relation between joint velocity of the robot  $\dot{\mathbf{q}} = [\dot{q}_1 \ \dot{q}_2 \ \dots \ \dot{q}_n]^T$  and its corresponding velocity in task space,  $\dot{\mathbf{r}} = [\dot{r}_1 \ \dot{r}_2 \ \dots \ \dot{r}_p]^T$ , is captured in terms of the robot Jacobian,  $\mathbf{J}_{rq}$ , as  $\dot{\mathbf{r}} = \mathbf{J}_{rq}\dot{\mathbf{q}}$ . The relation between feature velocities  $\dot{\mathbf{s}} = [\dot{s}_1 \ \dot{s}_2 \ \dots \ \dot{s}_m]^T$  and task space velocities, is given by  $\dot{\mathbf{s}} = \mathbf{J}_{sr}\dot{\mathbf{r}}$ . Thus, if the chosen feature is a point in the image, and the Cartesian coordinates of the camera are used, the relation is given by:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} f/Z & 0 & -u/Z & -uv/f & (f^2 + u^2)/f & -v \\ 0 & f/Z & -v/Z & -(f^2 + u^2)/f & uv/f & u \end{bmatrix} \begin{bmatrix} \mathbf{T} \\ \mathbf{W} \end{bmatrix} \tag{1}$$

where  $u, v$  represent the centered image coordinates,  $f$  is the focal distance,  $Z$  is the space coordinate and  $[\mathbf{T} \ \mathbf{W}]^T = [T_x \ T_y \ T_z \ w_x \ w_y \ w_z]^T$  are the components of the traslational and rotational speed of the point. By means of a transformation matrix we can change over to a task related coordinate system.

The velocity of the image features can be directly related to joint velocities in terms of a composite Jacobian, also known as the full visual-motor Jacobian [4], [5]:

$$\dot{\mathbf{s}} = \mathbf{J}_{sq}\dot{\mathbf{q}} = \begin{bmatrix} \frac{\partial s_1}{\partial q_1} & \dots & \frac{\partial s_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_m}{\partial q_1} & \dots & \frac{\partial s_m}{\partial q_n} \end{bmatrix} \dot{\mathbf{q}} \ ; \ \text{where } \mathbf{J}_{sq} = \mathbf{J}_{sr}\mathbf{J}_{rq} = \mathbf{J} \tag{2}$$

Determining this matrix analytically is not simple. It is necessary to remark that in its calculation there must be considered: the intrinsic parameters of the camera calibration (focal distance, image center coordinates), the 3D reconstruction of the point or an approximation ( $Z$  coordinate), the kinematic calibration of the camera (relation between camera coordinates and joint space origin), and the kinematic calibration of

the robot. Most of the previous works on visual servoing assume that the system structure and the system parameters are known, or the parameters can be identified in an off-line process. A control scheme with off-line parameter identification is not robust for disturbance, change of parameters, and unknown environments. One approach to image-based visual servoing without calibration is to dynamically estimate the full visual-motor Jacobian during motion.

**2.1 Estimation of the Jacobian**

Specialized literature gathers two methods for estimating the jacobian described in equation (3) [4], [5]. In both cases an initial jacobian is obtained by making n linearly independent small movements.

**2.1.1 Estimation Based on the Last Moves**

If the change in image features and the change in joint position are represented respectively by  $\Delta \mathbf{s}_k = \mathbf{s}_k - \mathbf{s}_{k-1}$  and by  $\Delta \mathbf{q}_k = \mathbf{q}_k - \mathbf{q}_{k-1}$ , and the image jacobian is assumed to be constant, it can then be estimated as the matrix that simultaneously satisfies  $n$  or more movements:

$$\begin{bmatrix} \Delta \mathbf{s}_{k-n+1}^T \\ \vdots \\ \Delta \mathbf{s}_k^T \end{bmatrix} = \begin{bmatrix} \Delta \mathbf{q}_{k-n+1}^T \\ \vdots \\ \Delta \mathbf{q}_k^T \end{bmatrix} \mathbf{J}^T \tag{3}$$

Once the jacobian has been obtained, the joint motion which allows approaching the desired features  $\mathbf{s}^*$  is calculated by:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (\mathbf{s}^* - \mathbf{s}_k) \tag{4}$$

**2.1.2 Recursive Estimation**

In this method the jacobian is estimated recursively, combining the information supplied by the last movement with the previous jacobian. Regarding the former method, it has the advantage of gathering information from all the movements. Applying the well-known Bryden method [6], [7]:

$$\mathbf{J}_{k+1} = \mathbf{J}_k + \frac{(\Delta \mathbf{s}_k - \mathbf{J}_k \Delta \mathbf{q}_k) \Delta \mathbf{q}_k^T}{\Delta \mathbf{q}_k^T \Delta \mathbf{q}_k} \tag{5}$$

**3 Proposed Algorithms**

The research line we have followed is the estimation of the jacobian based on already performed movements, not necessarily the  $n$  last ones. We use the visual information supplied by two cameras. We contribute two innovations: on one hand, each movement has been endowed with a certain reliability, so that the most adequate or reliable movements can be used. On the other hand, the epipolar constraint has been taken into account in the calculation of the jacobian, which significantly increases the robustness of the method, as will be seen in chapter four.

### 3.1 Reliability Estimation

The jacobian matrix, equation (1), strongly depends on the position of the point in the image  $(u, v)$ , so the assumption that it is constant will only be valid in the surroundings of the point in the image. For this reason, a first possibility would be to promote the consideration of those already performed movements with a short path in image features. However, these movements are too sensitive to noise, so an agreement must be reached between both effects. Movements performed in the joint surroundings of the desired movement also seem more adequate. The proposed algorithms rank the already performed movements according to a reliability which depends on two factors that assemble these concepts:

$$reliability_{ki} = Factor1_i / Factor2_{ki} \tag{6}$$

Where subindex k represents the last movement performed and subindex i will vary amongst those already performed movements which have been stored. We must remark that  $Factor1_i$  depends solely on already performed movements and promotes those in which features vary within a range: they are not too large so that the jacobian can be considered a constant (aprox. 20 pixels), nor too small so they will not be too sensitive to noise (aprox. 1 pixel).  $Factor2_{ki}$  also considers the last movement performed and promotes those already stored movements produced in the joint surroundings of the desired movement.

### 3.2 Adding the Epipolar Constraint

The projection of a point in two images satisfies an additional constraint known as the epipolar condition, [8], [9], expressed by the fundamental matrix (see equation (9)). The hereby method considers this constraint in the calculation of the image jacobian, equation (3). If the belonging of the variables to each of the cameras is denoted by inverted commas, we have the following model:

$$s_k = \begin{bmatrix} s'_k \\ s''_k \end{bmatrix} ; \quad s_{k-1} = \begin{bmatrix} s'_{k-1} \\ s''_{k-1} \end{bmatrix} ; \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}' \\ \mathbf{J}'' \end{bmatrix} \tag{7}$$

$$s'_k = s'_{k-1} + \mathbf{J}' \Delta \mathbf{q}_k \quad ; \quad s''_k = s''_{k-1} + \mathbf{J}'' \Delta \mathbf{q}_k \tag{8}$$

$$\begin{bmatrix} s''_{k-1} & 1 \end{bmatrix} \mathbf{F} \begin{bmatrix} s'_{k-1} \\ 1 \end{bmatrix} = 0 \quad ; \quad \begin{bmatrix} s''_k & 1 \end{bmatrix} \mathbf{F} \begin{bmatrix} s'_k \\ 1 \end{bmatrix} = 0 \tag{9}$$

Substituting in (9) the values obtained in (8), we have the following non-linear equation for  $(\mathbf{J}', \mathbf{J}'')$ :

$$\Delta \mathbf{q}_k^T \mathbf{J}''^T \mathbf{F} \mathbf{J}' \Delta \mathbf{q}_k + \Delta \mathbf{q}_k^T \mathbf{J}''^T \mathbf{F} \begin{bmatrix} s'_{k-1} \\ 1 \end{bmatrix} + \begin{bmatrix} s''_{k-1} & 1 \end{bmatrix} \mathbf{F} \mathbf{J}' \Delta \mathbf{q}_k = 0 \tag{10}$$

The linear equations in (3) and the non-linear equations in (10) have been jointly solved applying the Levenberg-Marquadt method.

### 3.3 Glossary of Proposed Algorithms

In the implementation described in the present article, the following algorithms have been employed depending on the movements used to calculate the image jacobian:

- Algorithm A: Last three movements performed
- Algorithm B: Three most reliable movements amongst the last ten performed.
- Algorithm C: Last ten movements performed, weighted by their reliability. Weighting is introduced multiplying each row in equation (4) by its corresponding reliability.
- Algorithm D: Ten most reliable movements performed.
- Algorithm E: Ten most reliable movements performed, adding the epipolar constraint.
- Algorithm F: Iterative estimation of the image jacobian.

Algorithms A, B, C, D, F are used for one or two cameras.

## 4 Experiments

In this section we describe our experimental equipment and results.

### 4.1 Experimental Equipment

The system used in the experiments consists of:

- A joint system composed of a high precision positioning device and its controller, model Newport MM3000 (see figure 1). The system has 3 DOF with a prismatic and two revolute joints, and its theoretical precision is of a thousandth of a millimeter and a thousandth of a degree. The visual control object, made out of five black dots on a white background, the projection of which on the image will be the control features, has been attached to the last link of the joint system.
- An image acquisition and processing system composed by two CV-M50 analogic cameras and a Matrox Meteor II-MC image acquisition board, which allows simultaneous acquisition from both cameras. The cameras, fixed in the working envi-

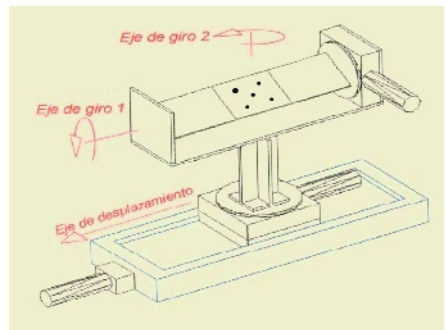
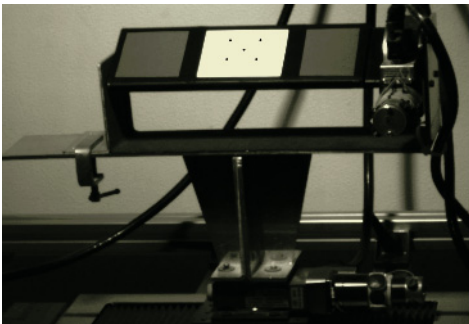


Fig. 1. Experimental equipment

ronment, are separated by about 300 millimeters, both their axes converge towards the joint system, and they are separated from it by about 700 millimeters. Visual features are detected with subpixel precision, and given the simplicity of the image, the error is estimated to be of less than 0.2 pixels. Communication with the joint system controller is established through a serial RS-232C cable.

## 4.2 Control Objective

The task entrusted to the system is to get the error between the desired and current visual features to be under a certain threshold. Visual features must be reachable and the visual object must be visible from both points of view. To ensure coherence, we decided to obtain the desired visual features previously by acquiring images in reference joint positions, chosen randomly within the workspace. To evaluate the effectiveness of each method, we consider four indices, defined as follows:

- Index 0: Sum of Euclidean distances between desired and current visual features. Weighted by number of points, number of cameras and number of trajectories.
- Index 1: Sum of Euclidean distances in joint space for all of the performed movements, divided by one thousand. Weighted by number of trajectories.
- Index 2: Average number of movements in which visual features are stabilised for each trajectory. We consider visual features to be stabilised when the error norm goes below a certain threshold. We used a 0.6 pixel threshold in our experiments.
- Index 3: Sum of Euclidean distances between desired and current visual features when overshooting occurs.

## 4.3 Results

A comparative study was conducted on the proposed algorithms. The comparison covers visual features calculated with an estimated error of 0.2 pixels, therefore considered without noise, as well as visual features with artificially added Gaussian noise with a standard deviation of 0.5 pixels. Also, the effect of increasing the number of points considered as visual features from 2 to 5 is analysed, as well as the effect of working with one or two cameras. Thus, table 1 gathers our results for the six proposed algorithms when using two cameras and analysing 50 trajectories covering the workspace, each with 30 movements. Without noise, the globally best-behaved algorithm is F, although others like D or E are also well-behaved. It is remarkable that incorporating reliability does not provide a noticeable advantage. When noise is added, algorithm F shows the worst behaviour, while D and E are well-behaved. Incorporating reliability does provide a significant advantage. In short, algorithm E has an outstanding immunity to noise, whilst algorithm F is very sensitive to it. Increasing the number of points does not imply improving the indices, as it makes a stricter control necessary. Table 2 gathers our results when using one camera for five of the proposed algorithms, for 50 trajectories, each with 30 movements, and without adding noise. The undeniable advantage of using two cameras can be appreciated. In figure 2 the evolution of algorithms F and E in the presence of noise is represented in joint space. Algorithm E is noticeably faster and more precise than algorithm F.

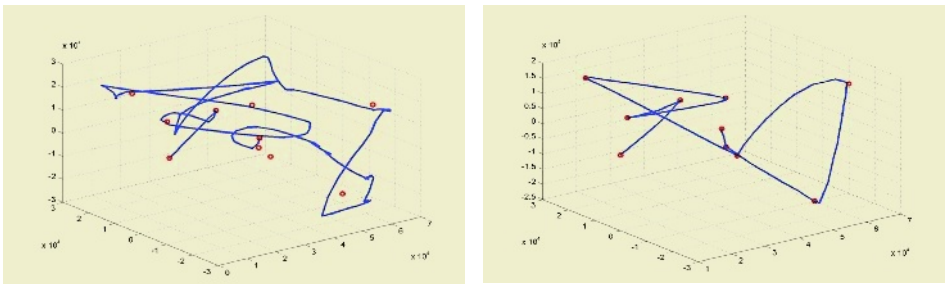


**Table 1.** Values for the four indices, with and without added noise

	ALGO-RITHM	WITHOUT NOISE				WITH NOISE			
		2 POINTS	3 POINTS	4 POINTS	5 POINTS	2 POINTS	3 POINTS	4 POINTS	5 POINTS
INDEX 0	A	316	328	298	305	696	644	695	731
	B	303	338	298	319	519	523	642	671
	C	273	285	279	285	348	420	432	467
	D	265	278	270	275	284	290	286	287
	E	263	275	270	277	264	281	277	278
	F	262	271	264	271	553	582	590	704
INDEX 1	A	44.1	43.8	43.6	41.5	90.3	77.6	79.3	70.9
	B	43.5	44.0	42.6	42.8	75.4	65.1	67.8	71.5
	C	41.0	40.3	39.3	39.0	54.2	52.3	53.4	54.5
	D	37.9	37.8	37.3	37.2	44.5	41.5	40.5	40.8
	E	38.0	37.5	37.3	37.3	41.7	39.8	40.4	39.9
	F	37.7	37.3	37.1	36.9	76.4	77.6	70.3	64.7
INDEX 2	A	11.7	11.4	13.5	13.2	29.4	29.8	29.8	29.9
	B	10.9	11.3	11.0	11.1	26.7	26.1	28.3	28.2
	C	10.1	10.1	9.8	10.1	18.4	20.4	24.0	25.5
	D	9.5	10.0	9.1	9.0	20.0	21.3	21.8	22.9
	E	9.4	10.0	9.1	9.0	15.2	15.5	17.2	21.4
	F	8.7	8.5	8.5	8.4	29.3	29.0	29.2	28.9
INDEX 3	A	1.7	1.8	3.7	1.7	30.5	31.9	30.9	33.8
	B	0.8	2.7	2.2	2.2	18.7	12.9	17.9	19.2
	C	1.4	1.3	1.1	1.4	6.8	4.7	4.4	7.6
	D	0.9	1.1	0.7	0.5	3.6	2.9	2.4	2.6
	E	1.6	0.8	0.7	0.5	2.3	2.7	3.8	3.1
	F	0.3	0.1	0.2	0.2	52.9	66.3	56.8	76.9

**Table 2.** Values for index 0 for one camera, without added noise

	ALGORITHM	2 POINTS	3 POINTS	4 POINTS	5 POINTS
INDEX 0	A	376	436	339	328
	B	369	443	342	357
	C	349	345	305	309
	D	298	335	296	308
	E	317	336	288	303



**Fig. 2.** Evolution of algorithms F and E in the presence of noise in joint space

## 5 Conclusions

The on-line estimation of the image jacobian is a flexible and versatile method for the visual control of a joint structure, since it isolates the obtained results from errors in

the calibration of the camera and the joint system. The paper contributes the definition of a reliability to calculate the jacobian, and the inclusion of the epipolar constraint or fundamental matrix in its calculation. This aspect is not considered in specialized literature, and it improves the results significantly when the noise level in feature detection increases. The knowledge of the fundamental matrix is no objection, as its calculation has been proven to be much more simple, robust and reliable than that of the complete calibration of the cameras and joint system.

Some aspects not dealt with in the present paper which are being currently studied are the analysis of the system stability with a control law generated from the jacobian estimation and the use of the proposed algorithms to accomplish dynamic tasks. Regarding the first aspect, we must remark that algorithm E has never made the system unstable throughout the many tests performed.

This work was supported by the Comisión Interministerial de Ciencia y Tecnología of the Spanish Government under the Project DPI2001-3827-C02-01.

## References

1. Corke, P.I: Visual Control of Robots: High Performance Visual Servoing. Research Studies Press (1996)
2. Hutchinson, S.A., Hager, G.D., Corke, P.I.: A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12-5 (1996) 651-670
3. Kragic, D. and Christensen: Survey on visual servoing for manipulation. Technical Report ISRN KTH/NA/P-02/01-Sen, CVAP259 (2002)
4. Deng, Z., Jägersand, M.: Evaluation of Model Independent Image-Based Visual Servoing, *Proceedings. First Canadian Conference on Computer and Robot Vision*,(2004) 138 - 144
5. Sutanto, H., Sharma, R., Varma, V.: The role of exploratory movement in visual servoing without calibration. *Robotics and Autonomous Systems* 23 (1998) 153-169
6. Piepmeier, J.A., McMurray, G.V., Lipkin, H.: Uncalibrated Dynamic Visual Servoing. *IEEE Transactions on Robotics and Automation*, vol 20, nº1 (2004) 143-147
7. Asada, M., Tanaka, T., Hosoda, K.: Adaptive Binocular Visual Servoing for Independently Moving Target Tracking. *International Conference on Robotics & Automation* (2000) 2076-2081
8. Hartley, R. Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge (2000)
9. Faugeras, O., Luong, Q.T.: *The Geometry of Multiple Images*. MIT Press (2001)

# Accelerometer Based Gesture Recognition Using Continuous HMMs

Timo Pylvänäinen

Nokia Research Center, P.O. Box 100, FIN-33721 Tampere  
timo.pylvanainen@nokia.com

**Abstract.** This paper presents a gesture recognition system based on continuous hidden Markov models. Gestures here are hand movements which are recorded by a 3D accelerometer embedded in a handheld device. In addition to standard hidden Markov model classifier, the recognition system has a preprocessing step which removes the effect of device orientation from the data. The performance of the recognizer is evaluated in both user dependent and user independent cases. The effects of sample resolution and sampling rate are studied in the user dependent case.

## 1 Introduction

Advances in microelectronics have reduced the cost of small and accurate sensors. As a result, these sensors are now finding their way to small mobile devices. This has inspired a lot of research effort around multi modal user interfaces.

Small and inexpensive accelerometers are now available. Already some mobile devices have such sensors embedded. These sensors can be used to record the movement of the device. It turns out, that simple gestures made by moving the device are relatively easy to recognize with high accuracy.

The basic tool for recognizing sequences of variable length is the Hidden Markov Model (HMM)[1]. HMMs have been successfully applied especially to speech recognition[2] and visual gesture recognition[4–6]. These methods translate almost directly to accelerometer-based gestures.

In this paper, a three dimensional accelerometer was used to record hand movements. Gestures, such as forming a circle or an upward line were recorded from 7 people. A recognizer based on continuous HMMs was implemented. The rest of the paper is organized as follows. In section 2 the preprocessing of the accelerometer data is discussed. In section 3 the HMM parameter estimation is described in some detail. Section 4 describes the experimental setup. Section 5 studies the effects of reducing sampling rate and sample resolution. And then finally section 6 summarizes the results.

## 2 Feature Extraction

In many recognition tasks it is crucial to extract some informative features from the recorded data. In speech recognition, this usually involves transformation to

frequency domain and decorrelation. One of the advantages of feature extraction is that it usually reduces the amount of information, both in terms of time resolution and in terms of dimensionality. Since the computational complexity of HMM decoding is linearly dependent on the number and dimension of feature vectors, good feature extraction also increases efficiency.

For accelerometer based gestures, however, the original data is naturally suited for HMMs. It is a plausible assumption that a gesture constitutes a sequence of accelerations to different directions. It is also very natural to assume that in practice, the recorded accelerations are naturally distributed around the expected direction.

Accelerometers that can be embedded in a handheld device must work without any outside reference. This usually means that the only observable quantity is force. A force on the sensor elements, however, is not necessarily due to actual acceleration. In particular, the gravitational pull of the Earth introduces a force that is interpreted as acceleration of considerable magnitude. If the gravitational constant  $\mathbf{g}$  is expected to be in the direction of the negative  $y$ -axis, then a tilt of only a few degrees will introduce an acceleration on  $z$  or  $x$ -axis comparable to accelerations due to motion of the device during a gesture. Thus a lot of information about the gesture comes from the tilting of the device during the gesture. This information is easily lost if the device is not held exactly in the same way for every repetition of the gesture.

These problems can be avoided by simple normalizing methods. Firstly, the effects of holding the device in a different manner can be somewhat canceled by rotating the data so that some estimate of  $\mathbf{g}$  based on the data is pointing to negative  $y$ -axis. Let

$$D = \begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_T^T \end{pmatrix},$$

where  $\mathbf{a}_i$  is the acceleration vector sampled at time  $i$ , be the acceleration data recorded from a gesture and  $\mathbf{g}_T(D)$  any linear mapping that estimates the direction of gravitational pull from the data. Then the normalizing problem is finding a matrix  $R$  such that  $\mathbf{g}_T(DR^T)^T = \alpha(0, -1, 0)$  with suitable constraints on  $R$ , where  $\alpha$  is a positive scalar value. By the linearity of  $\mathbf{g}_T$  this is equivalent to solving  $R\mathbf{g}(D) = \alpha(0, -1, 0)^T$ .

In general, this problem has uncountably many solutions, even with the obvious constraint that  $R$  should have full rank. In this paper the constraint is that  $R$  should be orthonormal, i.e.  $R^T R = I$ . This means that  $R$  represents a rotation or rotoinversion. Other constraints could be more suitable if, for instance, the sensor axes are not necessarily orthogonal. When the tilting is small, so that the angle between  $\mathbf{g}_T(D)$  and the negative  $y$ -axis is less than 90 degrees, the matrix  $R$  can be found as follows. Let

$$R = \begin{pmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{pmatrix} \quad (1)$$

and  $\mathbf{r}_2^T = -\frac{\mathbf{g}_T(D)}{|\mathbf{g}_T(D)|}$ . The small tilting condition implies that  $\mathbf{r}_2^T \hat{x}$  and  $\mathbf{r}_2^T \hat{z}$  are both nonzero, where  $\hat{x}$  and  $\hat{z}$  are the positive unit vectors in  $x$  and  $z$  directions respectively. Then complete  $R$  to a new base by using Gram-Schmidt orthogonalization (for example[3]) for  $\mathbf{r}_2$ ,  $\hat{x}$  and  $\hat{z}$ . More precisely, let

$$\mathbf{r}_1 = \frac{\hat{x} - \text{proj}_{\mathbf{r}_2}(\hat{x})}{|\hat{x} - \text{proj}_{\mathbf{r}_2}(\hat{x})|}, \quad (2)$$

$$\mathbf{r}_3' = \hat{z} - \text{proj}_{\mathbf{r}_2}(\hat{z}) \quad (3)$$

and

$$\mathbf{r}_3 = \frac{\mathbf{r}_3' - \text{proj}_{\mathbf{r}_1}(\mathbf{r}_3')}{|\mathbf{r}_3' - \text{proj}_{\mathbf{r}_1}(\mathbf{r}_3')|}. \quad (4)$$

It is easy to see that this solution is orthonormal when  $\mathbf{r}_2$  is not parallel with  $\hat{x}$  or  $\hat{z}$ .

One possible choice for  $\mathbf{g}_T$  is the mean of the acceleration vectors in  $D$ , i.e.  $\mathbf{g}_T(D) = \frac{1}{T}D^T \mathbf{1}_T$ , where  $\mathbf{1}_T$  is a vector of  $T$  ones. This choice was also used for the tests in this paper.

The state machine in HMM allows for variable duration of gestures. It is this fact, that makes HMM so useful for this type of pattern recognition task. In accelerometer based gestures, however, the magnitude of the observed accelerations also depend on the rate of the gesture. In contrast, HMM implicitly assumes that the state outputs are independent of the rate. Thus the data must be normalized to meet these assumptions.

The simplest way to do this is to scale by the factor  $\frac{1}{\max_t |\mathbf{a}_t|}$ . In the tests, the scaling improved recognition accuracy by a little over one percent unit.

### 3 Parameter Estimation

A left to right HMM with continuous normal output distributions was used. The output distributions were assumed to have diagonal covariance matrices. Thus the model can be described by  $8n$  parameters, where  $n$  is the number of states in the model. Each state has two transition probabilities and one Gaussian output distribution. For each state, the three dimensional output distribution is described by the mean vector  $\boldsymbol{\mu} \in \mathbb{R}^3$  and the three diagonal elements of the covariance matrix. Altogether eight values per state.

The  $8n$  parameters can be iteratively estimated from training gestures by the Baum-Welch algorithm[2]. The basic idea is to compute the probability  $\gamma_{ij}(t)$  of a transition from state  $i$  to state  $j$  at time  $t$  given that the model generated the given training gesture. Batch training was used, where the statistics  $\gamma_{ij}^{O_k}(t)$  were first computed for all training gestures  $O_k$ .

The improved estimate for the mean of state  $i$  is then

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_k \sum_{t=1}^{T_k} \sum_j \gamma_{ij}^{O_k}(t) O_k(t)}{\sum_k \sum_{t=1}^{T_k} \sum_j \gamma_{ij}^{O_k}(t)}. \quad (5)$$

Similarly, the estimate for the  $l$ :th diagonal element  $\sigma_i^2(l)$  of the covariance matrix for the output distribution of state  $i$  is

$$\hat{\sigma}_i^2(l) = \frac{\sum_k \sum_{t=1}^{T_k} \sum_j \gamma_{ij}^{O_k}(t) (\hat{\mu}_i(l) - O_{kl}(t))^2}{\sum_k \sum_{t=1}^{T_k} \sum_j \gamma_{ij}^{O_k}(t)} . \quad (6)$$

The estimate for the probability of transition from state  $i$  to state  $j$  is

$$\hat{a}_{ij} = \frac{\sum_k \sum_{t=1}^{T_k} \gamma_{ij}^{O_k}(t)}{\sum_k \sum_{t=1}^{T_k} \sum_j \gamma_{ij}^{O_k}(t)} . \quad (7)$$

The result by Baum and his colleagues guarantees that for these update formulas the combined probability of producing all the training gestures is increased or remains fixed. The parameter estimation was iterated three times for every model, initial parameters being zero mean, unit variance and fifty-fifty transition probabilities.

## 4 Experimental Setup

The performance of the recognizer was tested on a set of 10 gestures, 20 samples per gesture from 7 different persons, totaling 1400 gesture samples. Each model for all the 10 gestures had eight states. The data was normalized as described in Sect. 2, with constant scaling.

Two tests were conducted: user dependent and mixed user. For user dependent tests, three gestures samples were used for training per each person. The rest of the gestures from each person were used for testing. This process was repeated so that samples 1-3, 4-6, ... and 16-18 were used for training and the rest for testing. This cross-validation was used so that the choice of training vectors would not influence the results.

For 7 persons, 10 gestures, 6 repetitions, and 17 samples per repetition, a total of 7140 recognition operations were done. Of these, 6909 were successful, giving a 96.76 percent recognition accuracy.

For mixed user recognition, 3 samples from each person – a total of 21 samples – were used to train a single recognizer. Then all the remaining samples were recognized from every person with this recognizer. Again six runs were conducted, choosing consecutive samples for training and recognizing those that were not part of the training. Of the resulting 7140 recognition operations, 7123 were successful: a 99.76 percent recognition accuracy.

The fact that user independent performs better is unexpected. The most likely reason is that too few training vectors were used for user dependent and the models were over-fitted. On the other hand, mobile applications can hardly expect the user to repeat the gesture in training phase for more than two or three times. Since the focus is on mobile devices, only three training samples were used in the user dependent tests.

While such an accuracy seems extremely good, it is not entirely unexpected. Similar accuracy is common in speech recognition, for such a small set of possibilities. Increasing the number of gestures will undoubtedly reduce the accuracy. Gestures by hand movements, however, are not natural for humans. All the possible gestures have to be learned and so an extensive set of gestures becomes unpractical. For example, a user interface based on only ten gestures may already demand too much learning to be attractive for users.

## 5 Effects of Data Quality

Especially in consumer mobile devices, memory and processing power can be very limited. The number of bits per acceleration vector element affects the amount of memory required. The sampling rate of the accelerometer affects both memory and processing power.

Since the recognizer performed so well for the original data, it was tested how much the data could be quantized or decimated before recognition accuracy suffers. User dependent recognition test, as described in the previous section, were run on reduced quality data.

### 5.1 Effects of Quantization

The original data had 12 bits per sample for each axis. This was quantized to multiples of  $2^q$  for  $q = 0 \dots 12$ . In other words, to values representable by 12 to 0 bits respectively. The results are plotted in Fig. 1.

Part of the recognition error for low resolutions is due to pathological training situation. If the acceleration vectors from all the training samples that affect the output of some state are equal, the estimated variance becomes zero. The recognizer implementation was not designed to handle such pathological cases

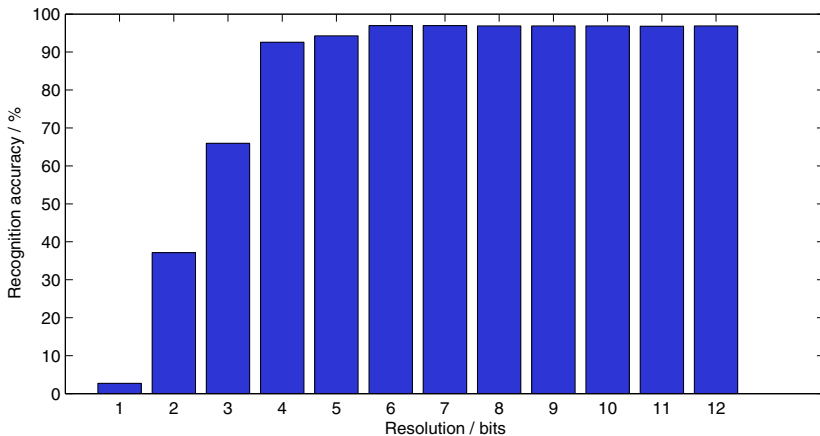


Fig. 1. Recognition accuracy as a function of sample resolution.

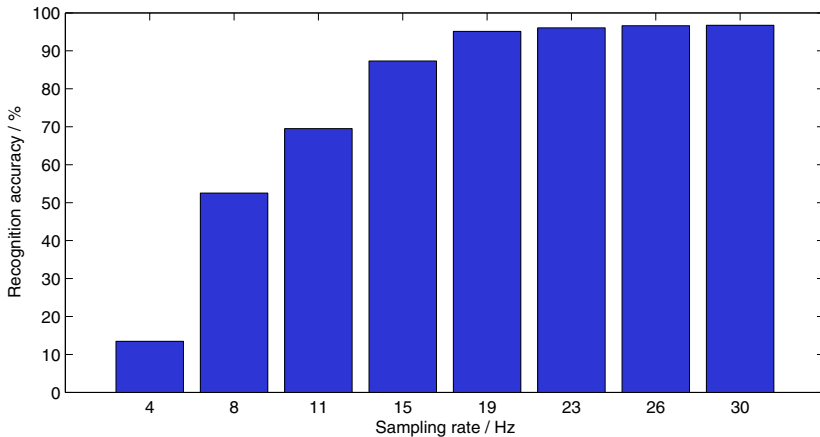
and the resulting model cannot be used for recognition. In such a case, all the samples of that gesture were interpreted as incorrectly recognized.

The results in Fig. 1 are expected. It appears that eight bits per axis is adequate for discriminating gestures. It is also conveniently a power of two and very common in commonly available microcontrollers.

## 5.2 Effects of Sampling Rate

The data was recorded on a mobile device running Symbian platform. The data polling was timed using Symbian methods and was subject to variability due to the multitasking nature of the operating system. Thus the actual sampling rate of the data is approximate and variable. It is, however, in the neighborhood of 30Hz. Certainly no more than 35Hz.

The original unquantized data was resampled to a frequency of  $\frac{q}{8}F_s$ ,  $q = 1..8$ , where  $F_s$  is the original sampling rate. Results are shown in Fig. 2. It can be seen, that the original sampling rate is closer to the critical limit than the resolution was.

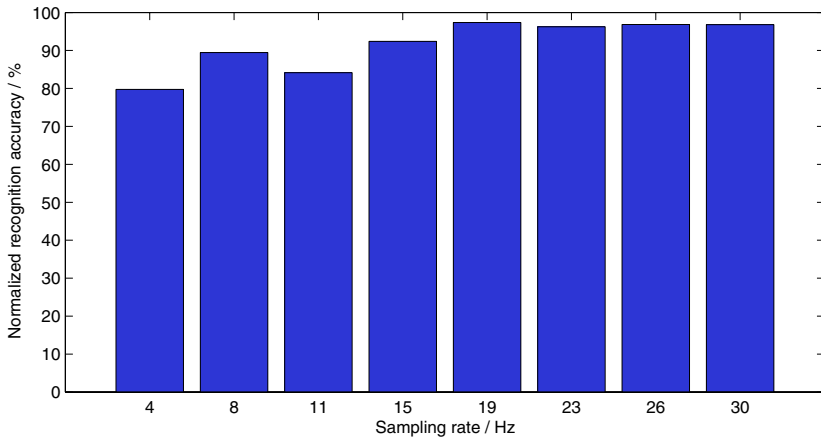


**Fig. 2.** Recognition accuracy as a function of sampling rate.

Notice that the eight state left to right HMM cannot decode a gesture of length less than eight. So for example for 8 Hz, gestures that are performed under a second become unrecognizable. In fact, this is the major source of recognition failures. Fig. 3 shows the recognition accuracy, when gestures that become too short are removed from the set for each sampling rate. The statistical significance of the results for small rates becomes weak, because for example for 4Hz there are only 236 gesture samples that can be used.

Figure 3 indicates that a major consideration in deciding the sampling rate is the expected duration and complexity of the gestures. Complex gestures require more states in the model, which in turn means that more samples per gesture must be collected.





**Fig. 3.** Percentage of gestures recognized, when too short gestures are removed from the set for each sampling rate.

## 6 Conclusion

A HMM based recognizer for accelerometer based gestures was implemented and tested. While actual feature extraction was not found necessary, some normalizing of the data was found beneficial. The effects of gesture sampling rate and quantization were studied.

The acceleration component due to gravitation plays an important role in the acceleration data resulting from a gesture. The direction of the gravitational component changes as a result of tilting the device. This occurs naturally, due to the shapes of the human hand and arm. The original orientation of the device in the palm can introduce a constant offset to this direction. A method of normalizing the data to compensate for this was described.

Standard HMM parameter estimation methods were tested and shown to produce good results. It was empirically shown that sampling rate does not seem to have a profound effect on the recognition accuracy, except when recorded gestures become too short to be recognized by the HMM. Eight bits per sample (per axis) was found to be sufficient resolution. Below 7 bits per sample, the recognition results start to noticeably deteriorate. These results are valid at least when the set of possible gestures has 10 gestures as in the tests. If, however, the gesture set is significantly larger, the gestures become more similar. In this case quantization in particular could have a more profound effect on the recognition accuracy.

## References

1. Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2001
2. Lawrence Rabiner, Biing-Hwang Juang, *Fundamentals Of Speech Recognition*, Prentice Hall, 1993

3. David Kincaid, Ward Cheney, *Numerical Analysis*, 2nd ed. Brooks/Cole, 1996
4. Stefan Eickeler, Adreas Kosmala, Gerhard Rigoll, *Hidden Markov Model Based Continuous Online Gesture Recognition*, In Proc. IEEE Int. Conf. on Pattern Recognition, 1998, pp. 1206 – 1208
5. Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, *Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 677 – 695, 1997
6. Matthew Brand, Nuria Oliver, Alex Pentland, *Coupled hidden Markov models for complex action recognition*, In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 994 – 999, 1997

# An Approach to Improve Online Hand-Eye Calibration

Fanhui Shi, Jianhua Wang, and Yuncai Liu

Inst. Image Processing & Pattern Recognition, Shanghai Jiao Tong University  
Shanghai 200030, P.R. China  
{fhshi, jian-hua.wang, whomliu}@sjtu.edu.cn

**Abstract.** Online implementation of robotic hand-eye calibration consists in determining the relative pose between the robot gripper/end-effector and the sensors mounted on it, as the robot makes unplanned movement. With noisy measurements, inevitable in real applications, the calibration is sensitive to small rotations. Moreover, degenerate cases such as pure translations are of no effect in hand-eye calibration. This paper proposes an algorithm of motion selection for hand-eye calibration. Using this method, not only can we avoid the degenerate cases, but also the small rotations to decrease the calibration error. Thus, the procedure lends itself to an online implementation of hand-eye calibration, where degenerate cases and small rotations frequently occur in the sampled motions. Simulation and real experiments validate our method.

## 1 Introduction

The calibration of robotic hand-eye relationship is a classical problem in robotics, which concerns the relative position and orientation between the robot gripper/end-effector and sensors, such as a camera mounted rigidly on the gripper. Hand-eye calibration is an important task for robot applications involving 3-D vision measurement, visual servoing, and tactile sensing.

On this problem, much work has been done by solving the homogeneous transformation equation  $\mathbf{AX}=\mathbf{XB}$  [1]-[7], which states that when the robot gripper undergoes a rigid motion  $\mathbf{A}$  and the corresponding camera motion is  $\mathbf{B}$ , the two motions are conjugated by the hand-eye transformation  $\mathbf{X}$ . Malm and Heyden [8] perform hand-eye calibration using normal derivatives of the image flow field instead of traditional point correspondences. And some methods simultaneously calibrate the camera and hand-eye relationship [9]-[11].

All the mentioned work requires an iterative approach, leading to offline least-square solutions. Angeles et al. [12] and Andreff et al. [13][14] first proposed the technique of online implementation of hand-eye calibration, allowing reducing human supervision required in classical calibration methods. Based on the linear invariants of rotation matrices, the former proposed a solution by recursive linear least squares. The latter derived a new linear formulation of the hand-eye problem, inspired by Sylvester equation:  $\mathbf{UV}+\mathbf{VW}=\mathbf{T}$ . Moreover, this method is extended to work with pose estimation by structure from motion. Therefore, it allows to get rid of the target objects required by standard approaches and use unknown scenes instead.

Note that whichever method is used, the hand-eye calibration problem intrinsically requires at least two motions with non-parallel rotation axes. This has been shown

algebraically [2] and geometrically [3]. So, hand-eye transformation from two independent motions sometimes can not be obtained when there exists a degenerate case such as pure translation or pure rotation etc., detailed algebraic analysis of the results for two independent motions refer to [14]. In addition, Tsai and Lenz made detailed analysis on critical factors affecting the accuracy of hand-eye calibration and got five observations, see [2].

During the online implementation of hand-eye calibration, hand-eye transformation is computed from continuously sampled motions, which are unplanned. So the problem is that, the sampled motions may be pure translation, pure rotation or their combinations, from which we only get partial calibration [14]. Furthermore, sampled motions may include a rotation with a tiny rotation angle, or the angle between two rotation axes is very small, both of which will bring on a large error in noisy measurements according to *observation 1* and *observation 2* in [2]. On the other side, according to *observation 4* in [2], small translation in gripper motion will be much useful in calibration.

In this paper we propose an algorithm of motion selection for online hand-eye calibration, which can not only avoid the degenerate cases in calibration, but also decrease the calibration error by selecting appropriate motion pairs. The remainder of this paper decomposes as follows. Section 2 describes the objective problem. Then, given golden rules for motion selection, the detailed algorithm of motion selection for online hand-eye calibration is presented in Section 3. Section 4 conducts some simulated and real experiments to validate the proposed algorithm.

## 2 Problem Formulation

We use upper-case boldface letters for matrices, e.g.  $\mathbf{X}$ , and lower-case boldface letters for 3-D vectors, e.g.  $\mathbf{x}$ . The angle between two vectors is denoted by  $\angle(\mathbf{x}, \mathbf{y})$ .  $\|\cdot\|$  means the Frobenius norm of a vector or a matrix. Rigid transformation is represented with a  $4 \times 4$  homogeneous matrix  $\mathbf{X}$ , which is often referred to as the couple  $(\mathbf{R}, \mathbf{t})$ . At the  $i$ -th measurement, the camera pose with respect to reference object is denoted by  $4 \times 4$  homogeneous matrix  $\mathbf{P}_i$ , and the recorded gripper pose relative to robot base is homogeneous matrix  $\mathbf{Q}_i$ .

The usual way to describe the hand-eye calibration is by means of homogeneous transformation matrices. We denote the transformation from gripper to camera by  $\mathbf{X}=(\mathbf{R}_x, \mathbf{t}_x)$ , the  $i$ -th motion matrix of the gripper by  $\mathbf{A}_i=(\mathbf{R}_{a,i}, \mathbf{t}_{a,i})$ , and the  $i$ -th motion matrix of the camera by  $\mathbf{B}_i=(\mathbf{R}_{b,i}, \mathbf{t}_{b,i})$ . The motion of the gripper is computed directly from the joint-angle readings by simple composition:

$$\mathbf{A}_i = \mathbf{Q}_i^{-1} \mathbf{Q}_{i+1} \quad (1)$$

With the known intrinsic camera parameters, the camera poses  $\mathbf{P}_i$  and  $\mathbf{P}_{i+1}$  relative to reference object are estimated, then the motion of the camera can also be determined by

$$\mathbf{B}_i = \mathbf{P}_i^{-1} \mathbf{P}_{i+1} \quad (2)$$

When dealing with an unknown scene (such as the scene without special calibration object), we can use a structure from motion algorithm [13][14] to estimate the camera

motion directly. Thus, the well-known hand-eye equation of  $A_i X = X B_i$  can be established [1][2], which yields one matrix and one vector equation:

$$R_{a,i} R_x = R_x R_{b,i} \tag{3}$$

$$(R_{a,i} - I) t_x = R_x t_{b,i} - t_{a,i} \tag{4}$$

As we know that two motions with non-parallel rotation axes are necessary to determine the hand-eye transformation, so another group of motion equations should be obtained,

$$R_{a,i+1} R_x = R_x R_{b,i+1} \tag{5}$$

$$(R_{a,i+1} - I) t_x = R_x t_{b,i+1} - t_{a,i+1} \tag{6}$$

Eqs. (3)-(6) can be combined into the following linear form [13][14]:

$$\begin{pmatrix} I_9 - R_{a,i} \otimes R_{b,i} & 0_{9 \times 3} \\ I_3 \otimes (t_{b,i})^T & I_3 - R_{a,i} \\ I_9 - R_{a,i+1} \otimes R_{b,i+1} & 0_{9 \times 3} \\ I_3 \otimes (t_{b,i+1})^T & I_3 - R_{a,i+1} \end{pmatrix} \begin{pmatrix} \text{vec}(R_x) \\ t_x \end{pmatrix} = \begin{pmatrix} 0_{9 \times 1} \\ t_{a,i} \\ 0_{9 \times 1} \\ t_{a,i+1} \end{pmatrix} \tag{7}$$

where the  $\otimes$  product is the Kronecker product and operator *vec* reorders (one line after the other) the elements of a  $m \times n$  matrix into a  $mn$  vector. Thus, given a pair of motions, hand-eye transformation can be linearly computed.

In practice, however, the movements of robot gripper vary with applications, not for hand-eye calibration. So, translations and small rotations usually occur in the sampled data, from which we can only get partial calibration or calibrate with big error. In order to make the online calibration practicable, we propose an approach of motion selection.

### 3 Motion Selection for Online Hand-Eye Calibration

We firstly give the golden rules for motion selection, and then describe two kinds of motion selection methods for online hand-eye calibration according to configurations.

#### 3.1 Golden Rules

As a rotation matrix  $R$  can be expressed as a rotation around a rotation axis  $k$  by an angle  $\theta$ , the relations between  $\theta$ ,  $k$  and  $R$  are given by Rodrigues theorem. Moreover,  $R_a$  and  $R_b$  have the same angle of rotation [1]. We can rewrite  $R_a$  and  $R_b$  as  $\text{Rot}(k_a, \theta)$  and  $\text{Rot}(k_b, \theta)$  respectively. Our aim in this paper is to sequentially find the pairs of consecutive motions  $(A_i, B_i)$  and  $(A_{i+1}, B_{i+1})$  for hand-eye computation by motion selection from sampled motion series. In this procedure, we should obey the following golden rules inferred from Tsai and Lenz's observations [2], that is

*Rule 1:* Try to make  $\angle(k_{a,i}, k_{a,i+1})$  (which is equal to  $\angle(k_{b,i}, k_{b,i+1})$  [2]) large, the minimal threshold is set to be  $\alpha$ .

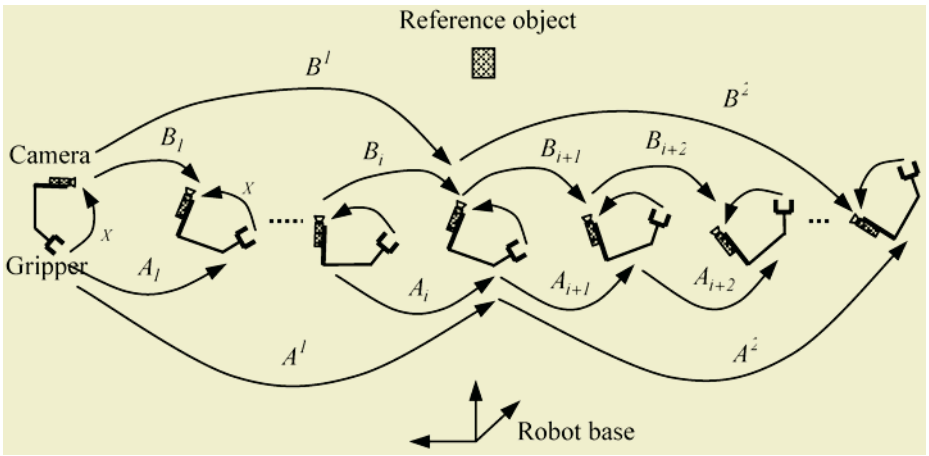
*Rule 2:* Try to make  $\theta_i$  large, the minimal threshold is  $\beta$ .

*Rule 3:* Try to make  $\|t_{a,i}\|$  small, the maximal threshold is  $d$ .

(As *observation* 3 and 5 in [2] are relative to the system configuration, we don't consider them in this paper). Thus, we can avoid degenerate motions and small rotations in solving hand-eye relationship to obtain higher calibration accuracy.

### 3.2 Motion Selection Algorithms

We denote the  $i$ -th sampled hand-eye pose and motion by  $(P_i, Q_i)$  and  $(A_i, B_i)$  respectively in this section.  $(\alpha, \beta, d)$  are threshold factors determined by experience and the detailed definitions are given in Section 3.1.  $(A', B')$  and  $(A'', B'')$  are selected motion pairs for calibration (see Fig. 1). For  $A'$  and  $A''$ , the rotation axis, rotation angle and translation are denoted by  $(k'_a, \theta', t'_a)$ ,  $(k''_a, \theta'', t''_a)$  respectively.



**Fig. 1.** Algorithm of motion selection for online hand-eye calibration.

Firstly, we consider the case when the camera pose  $P_i$  in each time instant can be estimated during the procedure.

At the beginning of the calibration process, we need to estimate  $(A', B')$ . The  $(A', B')$  is recovered from  $(P_1, Q_1)$  and  $(P_2, Q_2)$  according to Eqs. (1)-(2). If  $\theta' \geq \beta$  and  $\|t'_a\| \leq d$ , we claim that the  $(A', B')$  has been found. Or else, we continue to compute  $(A', B')$  from  $(P_1, Q_1)$  and  $(P_3, Q_3)$  and judge the value  $\theta'$  and  $\|t'_a\|$  in the same way as before. Repeat this procedure until  $\theta'$  and  $\|t'_a\|$  fulfill the given conditions. Here, we assume that the first  $(A', B')$  is estimated from  $(P_1, Q_1)$  and  $(P_i, Q_i)$ . After  $(A', B')$  has been found, another motion pair  $(A'', B'')$  can be sought starting from  $(P_i, Q_i)$  and  $(P_{i+1}, Q_{i+1})$  in the similar way as that of  $(A', B')$ , but the constrained conditions are changed to be  $\theta'' \geq \beta$ ,  $\|t''_a\| \leq d$  and  $\angle(k'_a, k''_a) \geq \alpha$ . When both motion pairs are found, we can make one calibration by solving Eq. (7).

In the next calibration, we take the last motion pair ( $A''$ ,  $B''$ ) as the new motion pair ( $A'$ ,  $B'$ ), and then continue to seek for new ( $A''$ ,  $B''$ ) from the successive sampled series and make a new hand-eye calibration in the same way as before.

We have thus derived an online hand-eye calibration algorithm based on iterative motion selection:

*Algorithm I*

1.  $i \leftarrow 2$ ;
2.  $A' = Q_1^{-1} Q_i$ ,  $B' = P_1^{-1} P_i$  ;
3. Compute  $\theta'$  and  $t'_a$  from  $A'$  ;
4. If  $\theta' \geq \beta$  and  $\|t'_a\| \leq d$ , then go to 6;
5.  $i \leftarrow i + 1$ , go to 2; (Sample one more motion)
6.  $j \leftarrow i + 1$ ; (Begin to search for  $A''$ )
7.  $A'' = Q_i^{-1} Q_j$ ,  $B'' = P_i^{-1} P_j$  ;
8. Compute  $\angle(k'_a, k''_a)$ ,  $\theta''$  and  $t''_a$  from  $A'$  and  $A''$  ;
9. If  $\angle(k'_a, k''_a) \geq \alpha$  and  $\theta'' \geq \beta$  and  $\|t''_a\| \leq d$ , then go to 11;
10.  $j \leftarrow j + 1$ , go to 7; (Sample one more motion)
11. Make one hand-eye calibration by solving Eq.(7);
12.  $A' \leftarrow A''$ ,  $B' \leftarrow B''$  ;
13.  $i \leftarrow j$ ,  $j \leftarrow j + 1$ , go to 7 for next calibration.

When dealing with an unknown scene, pose estimation has to be replaced by an algorithm of structure from motion. Therefore, instead of the camera motion  $B_i$  from pose estimation, a scaled motion  $\tilde{B}_i = (R_{b,i}, t_{b,i} / \|t_{b,i}\|)$  can be estimated, details refer to [13][14]. In this case,  $B'$  and  $B''$  can be computed by motion synthesis described in the following.

Considering two consecutive camera motions  $B_1$  and  $B_2$ , we can get the equivalent motion  $B$  as follows:

$$B = \begin{bmatrix} R_b & t_b \\ \theta^T & 1 \end{bmatrix} = \begin{bmatrix} R_{b,1} & t_{b,1} \\ \theta^T & 1 \end{bmatrix} \begin{bmatrix} R_{b,2} & t_{b,2} \\ \theta^T & 1 \end{bmatrix} \quad R_b = R_{b,1} R_{b,2}, t_b = R_{b,1} t_{b,2} + t_{b,1} \quad (9)$$

For convenience, we use symbol “ $\oplus$ ” in the following to denote the motion synthesis operation, thus Eq. (9) can be briefly rewritten as:

$$B = B_1 \oplus B_2 \quad (10)$$

The corresponding algorithm for online hand-eye calibration from unknown scenes is as follows:

*Algorithm II*

1.  $i \leftarrow 1$ ;
2.  $A' = Q_1^{-1} Q_2$ ,  $B' \leftarrow B_1$  ;

3. Compute  $\theta'$  and  $t'_a$  from  $A'$ ;
4. If  $\theta' \geq \beta$  and  $\|t'_a\| \leq d$ , then go to 7;  
else,  $i \leftarrow i + 1$ ; (Sample one more motion)
5.  $A' = Q_1^{-1} Q_{i+1}$ ,  $B' \leftarrow B' \oplus B_i$ , go to 3;
6.  $j \leftarrow i + 1$ ,  $i \leftarrow i + 1$ ; (Begin to search for  $A''$ )
7.  $A'' = Q_i^{-1} Q_{j+1}$ ,  $B'' \leftarrow B_j$ ;
8. Compute  $\angle(k'_a, k''_a)$ ,  $\theta''$  and  $t''_a$  from  $A'$  and  $A''$ ;
9. If  $\angle(k'_a, k''_a) \geq \alpha$  and  $\theta'' \geq \beta$  and  $\|t''_a\| \leq d$ , then go to 11;
10.  $j \leftarrow j + 1$ , go to 7; (Sample one more motion)
11. Make one hand-eye calibration by solving Eq.(7);
12.  $A' \leftarrow A''$ ,  $B' \leftarrow B''$ ;
13.  $i \leftarrow j + 1$ ,  $j \leftarrow j + 1$ , go to 7 for next calibration.

In *Algorithm I* and *Algorithm II*, motion pairs  $(A', B')$  and  $(A'', B'')$  are sought by an iterative procedure. To prevent the iteration from performing too many times, a threshold could be set to control the number of iterations.

## 4 Experiments

In this section, experiments on synthetic data and real scenes are carried out to validate our algorithm, where we adopt *Algorithm I*. To compare its performance, we make an additional experiment by directly solving Eq. (7) without any motion selection. In the following graphs, we denote the proposed method by “new method” and the direct approach by “traditional method”.

The motivation of the synthetic experiments is to test the performance of the new method by varying three threshold factors. The simulation is conducted as follows: we establish a consecutive motion series with 1000 hand stations  $Q_i$ . We add uniformly distributed random noise with relative amplitude of 0.1% on the rotation matrix and of 1% on the translation vector. We assume a hand-eye setup and compute the camera pose  $P_i$ , to which we also add uniformly distributed random noise as before.

For each factor, we calibrate the hand-eye relationship with different threshold while fixing the other two factors unchanged. In this way, we get the estimated rotation matrix  $\hat{R}$  and translation vector  $\hat{t}$ . To qualify the results, we take RMS of the errors in the rotation matrix and the RMS of the relative errors  $\|t - \hat{t}\| / \|t\|$  in the translation, which are customary error metrics in the literature [2][6][7]. Fig. 2 shows the simulation results. On one hand, the motion selection approach exhibits better behavior than the method without motion selection when there exist noisy measurements. The characteristics of error varying with different threshold validate the rules in Section 3.1.

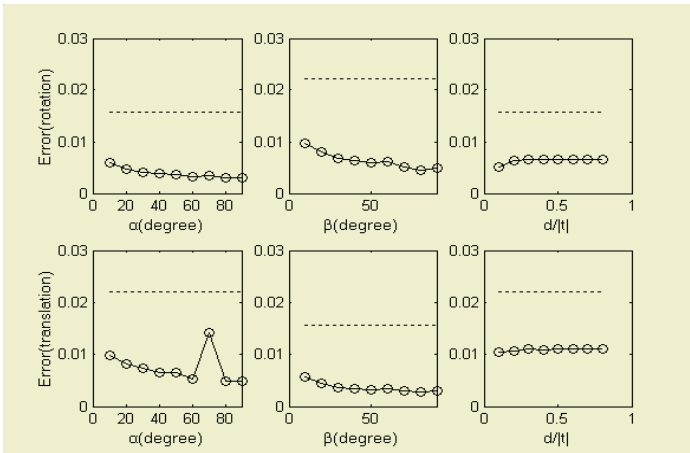
From the results of the experiment, we can find that the impact of variation of  $d$  on the error is unnoticeable. So, in the second experiment, we test the behavior of the new algorithm with fixed  $d$  and simultaneous variation in  $(\alpha, \beta)$ . Fig. 3 shows the result of



the second test, from which we can see that the calibration error is hard to be decreased when factors ( $\alpha, \beta$ ) are great than  $30^\circ$ . Note that in practice, when the two threshold increase, the average number of motions in each selection will increase, which will decrease the performance of system in real-time applications.

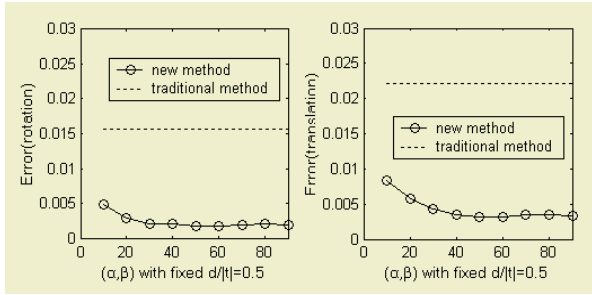
We also demonstrate the foregoing algorithm on a real setup composed of an infrared marker and a pair of CCD cameras which are attached to the end-effector of a 6-DOF robot (MOTOMAN CYR-UPJ3-B00), see Fig. 4(a). After the stereo rig is precisely calibrated, we mount an infrared filter on each camera. Thus, we get an infrared navigation system with stereoscopic vision. Without loss of generality, we compute the hand-eye transformation between the left camera and the gripper.

In the test, the robot is fixed on a workbench and the moving cameras observe the static infrared mark. We randomly move the gripper to 24 locations with different relative rotation or/and translation controlled by program; repeat the test for 10 times. For each time instant, gripper pose  $Q_i$  can be read from robot controller and pose of reference object  $P_i$  relative to the camera can be solved by binocular vision. We perform the hand-eye calibration using the same methods as in the synthetic experiments. In the test with motion selection, the values of three factors need to be set by experience at first.



**Fig. 2.** Performances of the new algorithm with variation in  $\alpha, \beta$  or  $d$ , which are compared to the traditional method without motion selection. The RMS rotation error is shown on the top and the RMS relative translation error is shown on the bottom, where the solid with lable “o” denote new method and the dotted denote traditional method.

As no ground-truth value is available for comparison in such experiments, we compared  $A_i X$  and  $X B_i$  for each motion  $i$ , and then gathered all these errors into RMS errors. This kind of measurement was also adopted by Andreff et al.[13]. The results of the experiments are shown in Fig 4(b), where the calibration times of traditional method does not include degenerate case. From Fig 4(b) we can see that, the average RMS error of proposed method is much lower than that of the traditional method.



**Fig. 3.** Performance of the new algorithm with fixed  $d$  and simultaneous variation in  $(\alpha, \beta)$ , which is compared to the traditional method without motion selection. The RMS rotation error is shown on the left and the RMS relative translation error is on the right.

### 5 Conclusion and Open Issue

In this paper, we propose an algorithm of motion selection for online hand-eye calibration, which can not only avoid the degenerate cases in hand-eye calibration, but also try to decrease the calibration error by selecting appropriate motion pairs. Experimental results from simulated data and real setup show that the method can greatly decrease the error of online hand-eye calibration.



(a)

	Average stations	Average times of calibration	Average RMS Error
Traditional method	24	12.5	1.1377
New method	24	3.3	0.6523

(b)

**Fig. 4.** Real experiment for online hand-eye calibration. (a). Experimental system. (b) Results of the real experiment.

However, the characteristics of gripper motion in different applications are diverse, so the threshold of the three factors should be chosen by experience according to the real setup. Open issue includes the following problem: “What is the optimal motion selection for online hand-eye calibration?”

### References

1. Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form  $AX = XB$ ," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 16-29, Feb. 1989.
2. R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration", *IEEE Trans. Robot. Automat.*, vol. 5, pp. 345-358, 1989.

3. H. Chen. "A screw motion approach to uniqueness analysis of head-eye geometry". in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Maui, Hawaii, USA, pp. 145-151, June 1991.
4. C. Wang. "Extrinsic calibration of a robot sensor mounted on a robot". *IEEE Trans. Robot. Automat.*, 8(2):161-175, Apr. 1992.
5. Hanqi Zhuang and Yui Cheung Shiu, "A Noise-Tolerant Algorithm for Robotic Hand-Eye Calibration With or Without Sensor Orientation Measurement". *IEEE Trans. on System, Man and Cybernetics*, 23(4):1168-1175,1993.
6. R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, 14(3):195–210, 1995.
7. K. Daniilidis. "Hand-eye calibration using dual quaternions". *Int. J. Robot. Res.*, 18(3):286-298, 1999.
8. H.Malm, A Heyden,. "A new approach to hand-eye calibration", in *Proc. 15th Int. Conf. Pattern Recognition*, Vol. 1 , pp. 525-529, Sept. 2000.
9. S. Ma, "A self-calibration technique for active vision systems", *IEEE Trans. Robot. Automat.*, 12(1):114-120, Feb. 1996.
10. G. Wei, K. Arbter, and G. Hirzinger. "Active self-calibration of robotic eyes and hand-eye relationships with model identification". *IEEE Trans. Robot. Automat.*, 14(1):158-166, 1998.
11. H. Malm, and A. Heyden, "Simplified intrinsic camera calibration and hand-eye calibration for robot vision", in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vol.1, pp. 1037-1043, Oct.2003.
12. J. Angeles, G. Soucy and F. P. Ferrie, "The online solution of the hand-eye problem", *IEEE Trans. Robot. Automat.*, vol. 16, pp. 720-731, Dec. 2000.
13. N. Andreff, R. Horaud and B. Espiau, "On-line hand-eye calibration", in *Proc. Int. Conf. on 3-D Digital Imaging and Modeling*, pp. 430 - 436, Oct. 1999.
14. N. Andreff, R. Horaud, and B. Espiau, "Robot hand-eye calibration using structure-from-motion", *Int. J. Robot. Res.*, 20(3):228–248, 2001.

## Part IX

# Hardware Architectures

# Image Processing Application Development: From Rapid Prototyping to SW/HW Co-simulation and Automated Code Generation

Cristina Vicente-Chicote<sup>1</sup>, Ana Toledo<sup>2</sup>, and Pedro Sánchez-Palma<sup>1</sup>

<sup>1</sup> Departamento de Tecnologías de la Información y Comunicaciones  
E.T.S. Ingeniería de Telecomunicación - Universidad Politécnica de Cartagena  
Campus Muralla del Mar S/N, 30.202 Cartagena, Spain  
{Cristina.Vicente, Pedro.Sanchez}@upct.es

<sup>2</sup> Departamento de Tecnología Electrónica  
E.T.S. Ingeniería Industrial - Universidad Politécnica de Cartagena  
Campus Muralla del Mar S/N, 30.202 Cartagena, Spain  
Ana.Toledo@upct.es

**Abstract.** Nowadays, the market-place offers quite powerful and low cost re-configurable hardware devices and a wide range of software tools which find application in the image processing field. However, most of the image processing application designs and their latter deployment on specific hardware devices is still carried out quite costly by hand. This paper presents a new approach to image processing application development, which tackles the historic question of how filling the gap existing between rapid throwaway software designs and final software/hardware implementations. A new graphical component-based tool has been implemented which allows to comprehensively develop this kind of applications, from functional and architectural prototyping stages to software/hardware co-simulation and final code generation. Building this tool has been possible thanks to the synergy that arises from the integration of several of the pre-existent software and hardware image processing libraries and tools.

**Keywords:** image processing applications, component-based development, prototyping, co-simulation, automated code generation

## 1 Introduction

Today, Image Processing (IP) techniques find application in many different domains such as automated visual inspection of industrial products, medical imaging or biometric person authentication, among others [1][2]. The marketplace offers many IP-related products, ranging from platform-optimized software and hardware libraries to high-level prototyping and simulation tools. Nevertheless, none of these products actually covers the whole process of building IP applications. Actually, the historic question of how bridging the gap between design models and final system implementation remains still open, also when talking about these systems.

This paper presents a novel approach to IP application development which is aimed to cover the whole life cycle of this kind of products. In order to put this approach into practice, a new tool has been implemented which, following the growing trend toward component-based application development [3], integrates some of the previously existing IP-related products, instead of being built from scratch.

The rest of this paper is organized as follows. The common procedure followed to build IP applications is briefly reviewed in Section 2. In Section 3, a new IP Comprehensive Development (IP-CoDe) Tool is presented, which is intended to help building and evaluating both functional and architectural IP prototypes. The use of this tool to develop a complete study case is presented in section 4. Finally, some conclusions and future research lines are included in Section 5.

## 2 Building IP Applications

Building IP applications usually requires an initial rapid prototyping stage which helps selecting the algorithms that fulfill the system functional requirements. Commonly, this functional prototype is implemented by means of a high level programming language (C++, MATLAB, Java, etc), and generally incorporates the functionality provided by one of the multiple available IP libraries, e.g. Intel© Open Computer Vision (free Open Source library) [4], Intel© Integrated Performance Primitives [5], Matrox© Imaging Library [6], Mathworks© IP Toolbox [7], etc.

Once the functional prototype has been carefully tested, the application architecture must be defined in terms of a specific platform which might be composed of several processors, whether SW or HW, or both. Thus, the initial prototype is partitioned into functional units that can be mapped into the different processing elements. This architectural design stage produces a co-prototype which must be tested in order to ensure that cost and performance constraints are met for each particular application. Testing the selected co-prototype is usually accomplished by means of co-simulation techniques, which allow evaluating both software and hardware, and their interactions (synchronization, data transfer, etc).

Thus, building IP applications requires a great deal of IP algorithms and configurations to be explored. In fact, different functional prototypes can fulfill the initial IP requirements. For each prototype, different SW/HW partitions can be obtained and, for each partition, different mappings of its functional units into the various elements of the platform can be selected. Finally, different platforms can be considered candidates for a given application.

Each of these design tasks can be developed by means of different tools, but as stated in [8] *“a new generation of tool is required which helps bridging the gap between the exiting design tools. Such tool, should address the functional and architectural design stages, and reach both the software and hardware domains”*.

## 3 The IP-CoDe Tool: An Integration Experience

Prototyping is a rapid and inexpensive way to validate system requirements. Usually, different prototypes are built in order to test different aspects of the application under

development. As a matter of fact, functional models are built as software throwaway prototypes by means of specialized tools, different from those needed for architectural co-prototyping, where HW devices must be also taken into account. Integrating these tools under a unified environment would ease evolving functional prototypes to the corresponding co-prototypes, thus filling the existing gap between application design and implementation.

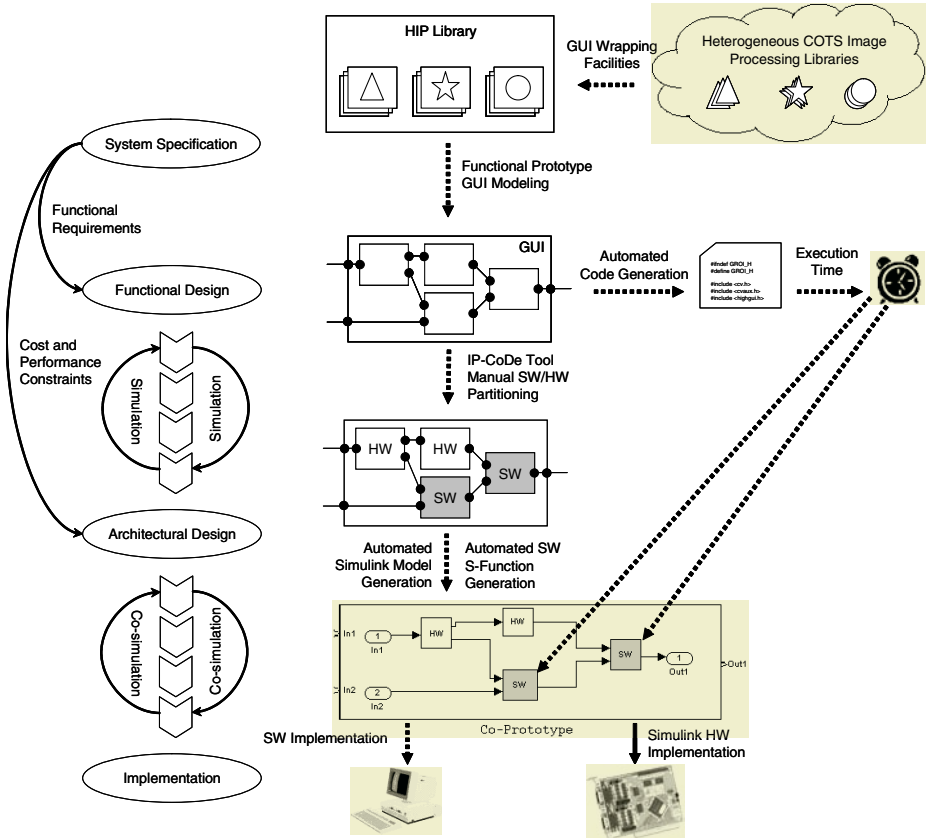


Fig. 1. Scheme of the IP application development life cycle using the IP-CoDe Tool.

In the following sections, our experience with IP product integration to build an IP Comprehensive Development Tool is detailed. This tool covers the whole IP application development life cycle, as shown in Fig. 1.

### 3.1 Functional Design

The first stage when building any application is to define its functional and non-functional requirements. IP application functional requirements typically deal with the selection of the algorithms that must be applied to input images in order to extract

relevant visual features (color, shape, texture, etc), while non-functional requirements are commonly related to cost, synchronization and timing issues.

As stated in section 2, the functional modeling stage is usually accomplished by means of one of the multiple IP libraries available. Although these libraries are functionally overlapped to some extent, they cover different aspects and consequently, they can be considered complementary. Thus, being able to simultaneously employ a mixture of them for building functional prototypes would be both useful and enriching. However, each IP library uses its own defined data structures, function calling conventions and error handling mechanisms, making it difficult to join them together.

In order to integrate the functionality provided by several of the existing IP libraries and toolboxes [4-7], the IP-CoDe Tool provides a wrapping mechanism which allows building homogeneous and inter-connectable IP components from heterogeneous IP functions. Wrappers allow mapping data representations, adding functionality to (or masking unneeded functionality of) components, and provide a higher level of abstraction to the components [9][10].

The IP-CoDe Tool provides a template for homogeneous IP component generation. In order to fill in this template, the user must provide (1) the signature of the function being wrapped, i.e. the number and type of its parameters, (2) the external interface of the component, i.e. the number and type of its connectors, and (3) the function that links each component connector to one of the function parameters. Once this template has been filled in, the IP-CoDe Tool automatically generates the corresponding wrapper, and thus a new component which is added to a repository of homogeneous and inter-connectable IP components for its latter use.

The IP-Code Tool allows building functional prototypes in a very rapid and intuitive way due to its Graphical User Interface (GUI), which makes it possible to “drag and drop” and interconnect any number of components selected from the repository. Any functional model depicted using this tool may be wrapped as well, in order to build a new higher-level functional component.

When the depicted functional prototype seems to be complete, the user can automatically obtain the corresponding code, which can be compiled and linked in order to produce a running prototype that allow testing the functional behavior using different input data (see Fig. 1).

### 3.2 Architectural Design

Building an architectural co-prototype implies selecting a specific platform on which to deploy the functional prototype. Thus, at this stage a SW/HW partitioning must be decided and non-functional requirements must be tested. As mentioned in section 2, these tests require a co-simulation tool.

Among the existing simulation tools, Simulink [11] is one of the most popular, mainly owing to its straight forward connection to MATLAB and to its graphical easy-to-use interface. Actually, Simulink can be used as a co-simulation tool, as both SW and HW blocks can be incorporated as a part of the system under simulation. SW blocks can be obtained from the many existing Simulink Toolboxes, and can also encapsulate MATLAB, C/C++, FORTRAN and Ada functions (S-functions). HW



blocks can be obtained from the various HW device-specific Toolboxes (e.g. System Generator [12] for the Xilinx FPGA<sup>1</sup>). These are some of the reasons why Simulink has been selected as the co-simulation tool for the IP-CoDe Tool.

After a SW/HW partitioning of the functional prototype has been decided (manually so far), the IP-CoDe Tool automatically deploys the corresponding Simulink co-prototype (mdl file). Each SW block in this co-prototype is then automatically filled in with an S-function automatically built by the IP-CoDe Tool from the corresponding component in the functional prototype. The estimated execution time of each SW block is then calculated, as this information is required for timing and synchronization purposes during the co-simulation (see Fig. 1).

In order to fill in the HW blocks, a library of IP high-level HW components has been created from a set of low-level functions included in the System Generator Toolbox for Simulink [11]. Some other HW blocks have also been included in this library directly from VHDL<sup>2</sup>-cores using a wrapper mechanism to allow their interconnection with the former ones. It is worth noting that all these HW blocks can be directly simulated by Simulink<sup>3</sup> without needing to buy any specific HW device. However, Simulink can also be used for generating and transferring the corresponding VHDL code for each HW block to a target FPGA in order to speed up the co-simulation.

Once the co-prototype is finished, co-simulation allows checking the non-functional requirement fulfillment. For instance, the execution time information provided by the co-simulation allows checking whether the synchronization and timing requirements are met. In the same way, other low-level HW requirements (e.g. maximum working frequency or FPGA area occupancy), can also be retrieved and checked.

### 3.3 Implementation

After carefully testing and fine-tuning the co-prototype, the final code of the IP application can be automatically obtained. Actually, the code associated to each software block is obtained during the functional prototyping stage, and the VHDL code corresponding to the HW blocks is straightly obtained by the System Generator Toolbox.

## 4 A Practical Study Case: Detecting Contours in Skin Regions

In order to test the IP-CoDe Tool, a complete IP application has been developed which allows detecting contours in human skin regions contained in color images.

Firstly, a functional prototype of the study case application was graphically built using the depicting and interconnection facilities provided by the IP-Code Tool GUI.

---

<sup>1</sup> FPGA stands for Field Programmable Gate Array.

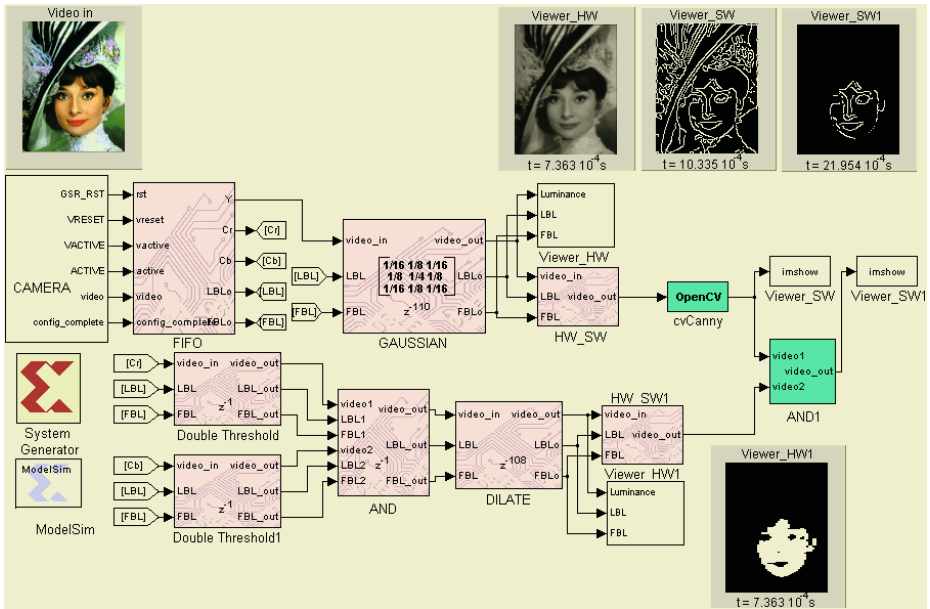
<sup>2</sup> VHDL stands for VHSIC (Very High-Speed Integrated Circuit) Hardware Design Language.

<sup>3</sup> System Generator blocks are supplied with a functionally equivalent software Simulink block that can be used for simulation. On the other hand, HW blocks directly obtained from VHDL-cores can also be simulated in Simulink using an external tool, e.g. ModelSim [13].

When the prototype seemed to be finished it was compiled and the corresponding executable version was automatically generated. However, after testing this prototype using several input images, some errors were detected and thus, the prototype had to be modified, fine-tuned and recompiled and again.

Once the functional prototype had been tested and the execution time associated to each component had been measured, the corresponding architectural co-prototype was built by manually selecting which components should be implemented in HW and which ones in SW, thus allowing the Simulink model (mdl file) to be automatically generated and co-simulated.

It is worth noting that, despite the changes and adjustments introduced during the functional prototyping, completing this stage took just a few minutes. On the other hand, it also should be noticed that despite the small size of the images employed, the co-simulation took about fifteen minutes to complete. In the case of full-sized images, the co-simulation time should be measured in hours. This leads to the conclusion that changes introduced at the architectural level have a much greater impact in the deploying time than those performed at the functional level.

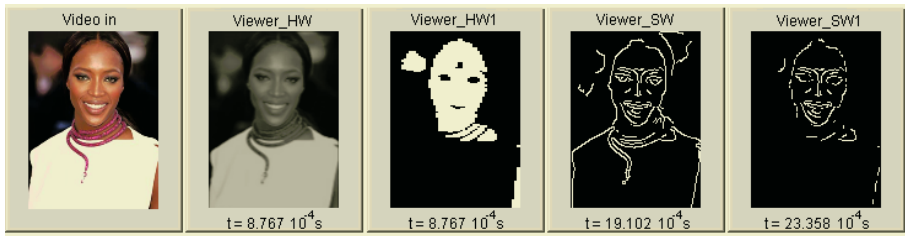


**Fig. 2.** Simulink co-simulation screenshot. SW blocks are shown in a plain-color while patterned ones denote HW components. Images resulting from each SW/HW processing step are shown together with the corresponding temporization.

The final co-prototype and the co-simulation results are shown in Fig. 2 where different kinds of blocks are shown: patterned blocks represent HW components while plain-color ones correspond to SW elements. HW blocks were obtained from two different sources: System Generator Simulink Toolbox (Xilinx), and a wrapped

VHDL-core obtained from Nallatech [14]. Similarly, SW blocks were obtained from Matlab C/C++ and Intel OpenCV functions. White blocks represent components needed for the simulation (visualization probes or external elements, e.g. a camera model) but that will not have any code associated in the final implementation.

Fig. 2 shows the images resulting from every SW/HW processing step. These images show the instant when they have been generated, thus allowing estimating the temporization of the final implementation. Fig. 3 shows the results obtained by the co-prototype using a different input image. This example proves that the designed application is robust to different skin colors.



**Fig. 3.** Results obtained using a different input image. From left to right: original image, smoothed intensity component, skin mask obtained by applying a threshold to the chroma components, Canny contours [15], and logical AND applied to the two previous images.

## 5 Conclusions and Future Research

This paper presents a new approach to IP application development that covers from functional and architectural prototyping stages to SW/HW co-simulation and final code generation. Building such a comprehensive tool has been possible thanks to the synergy that arises from the integration of several preexistent IP-related products. A complete IP application for contour detection in human skin regions has been wholly developed using the IP-CoDe Tool as a study case.

At present, the IP-CoDe Tool only allows building feed-forward functional prototypes. Extending this functionality to allow the presence of loops will widen the range of applications that could be created. It would also be interesting to find new tools, which being integrated with the existing ones could help automating the SW/HW partitioning to some extent, finding bottlenecks, or detecting which parts of the generated code are more susceptible of being parallelized.

## Acknowledgements

This work has been partially supported by the Spanish CITYT Project COSIVA (TIC 2000-1765-C03-02), the European Community Project EFTCOR (DPI2002-11583-E), and the mobility program for researchers of the Technical University of Cartagena (PMPDI-UPCT-2004). We would also like to thank Dr. M. Pinzolas-Prado for his help and support, and A. J. Martínez-Lamberto and J. A. Martínez-Navarro for their collaboration in the IP-CoDe Tool implementation.

## References

1. Bovik, A.: Handbook of Image and Video Processing, Academic Press (2000) 749-869.
2. Vicente-Chicote, C., Fernández-Andrés, C., Sánchez-Palma, P.: Automated Visual Inspection Systems Development from a Generic Architectural Pattern Description (in Spanish), NOVATICA, Vol. 171, (2004) 63-65.
3. Bass, L., et al.: Volume II: Technical Concepts of Component-Based Software Engineering, SEI Technical Report CMU/SEI-2000-TR-008. May 2000.
4. Intel® OpenCV. Available: <http://www.intel.com/research/mrl/research/opencv>
5. Intel® IPP. Available: <http://www.intel.com/software/products/ipp/>
6. Matrox® MIL version 7.5. Available: <http://www.matrox.com/imaging/products/mil>
7. The Mathworks® Image Processing Toolbox 5.  
Available: <http://www.mathworks.com/products/image/>
8. Perrier, V.: A look inside Electronic System Level (ESL) design, CMP United Business Media, EEDesign.com, Article Id. 18402916, March 26 (2004).
9. Dean, J.C., Vigder, M.R.: System Implementation Using Commercial Off-The-Shelf Software, National Research Council Canada (NCR), Report 40173 (1997).
10. Troya, J. M., Vallecillo, A.: Controllers: Reusable Wrappers to Adapt Software Components. Information & Software Technology, Vol. 43(3), (2001) 189-202.
11. Simulink® 6. Available: <http://www.mathworks.com/products/simulink/>
12. System Generator. Available: [www.xilinx.com/products/design\\_resources/design\\_tool](http://www.xilinx.com/products/design_resources/design_tool)
13. ModelSim. Available: [www.model.com](http://www.model.com)
14. Nallatec sample IP VHDL-core. Available: [www.Nallatech.com](http://www.Nallatech.com)
15. Canny, J.F.: A Computational Approach to Edge Detection. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 8 No. 6 (1986) 679-698.

# Xilinx System Generator Based HW Components for Rapid Prototyping of Computer Vision SW/HW Systems\*

Ana Toledo<sup>1</sup>, Cristina Vicente-Chicote<sup>2</sup>, Juan Suardíaz<sup>1</sup>, and Sergio Cuenca<sup>3</sup>

<sup>1</sup> Departamento de Tecnología Electrónica, Universidad Politécnica de Cartagena, Spain  
{ana.toledo,juan.suardiaz}@upct.es

<sup>2</sup> Departamento de Tecnologías de la Información y Comunicaciones  
Universidad Politécnica de Cartagena, Spain  
cristina.vicente@upct.es

<sup>3</sup> Departamento de Tecnología Informática y Computación, Universidad de Alicante, Spain  
sergio@dtic.ua.es

**Abstract.** This paper shows how the Xilinx System Generator can be used to develop hardware-based computer vision algorithms from a system level approach without the necessity of in-depth knowing neither a hardware description language nor the particulars of the hardware platform. Also, it is demonstrated that Simulink can be employed as a co-design and co-simulation platform for rapid prototyping of Computer Vision HW/SW systems. To do this, a library of optimized image processing components based on XSG and Matlab has been developed and tested in hybrid schemes including HW and SW modules. As a part of the testing, results of the prototyping and co-simulation of a HW/SW Computer Vision System for the automated inspection of tangerine segments are presented.

**Keywords:** image processing applications, FPGAs, prototyping, co-simulation, Simulink.

## 1 Introduction

Nowadays, the key for implementing high-performance digital signal processing (DSP) systems, especially in digital communications, video and image processing applications, is the use of programmable logical devices, in particular Field Programmable Gate Arrays (FPGAs). However, for those applications in which high-level complex algorithms are involved, a complete HW implementation is unpractical. In these cases it is usual to employ a hybrid SW/HW implementation, in which the hardware (typically a FPGA) carries out the acceleration of specialized functions and a processor, usually a conventional CPU, accomplishes general purpose computing.

Traditionally, the HW/SW application prototyping is performed in different environments, using a high-level programming language for the SW, e.g. C, C++ or Mat-

---

\* This work has been partially supported by the Spanish CITYT Project COSIVA (TIC 2000-1765-C03-02).

lab, and a Hardware Description Language (VHDL or Verilog) for the HW description. This makes the co-simulation difficult, and leads to two versions of the same code in different programming languages. Moreover, the rapid evolution of FPGAs makes the time employed in the development of hardware-based applications be a critical parameter. For these reasons, some efforts have been made to construct co-design environments allowing rapid prototyping, high-level modeling, co-simulation, and straightforward HW/SW code generation [1-3].

Currently, there exist two main tendencies in the system-level co-design environments: high-level languages [4] and dataflow-based visual environments [5, 6]. High level languages are efficient for specification modelling and algorithm verification, but they are not suitable for the implementation of high-performance dataflow systems as in the Computer Vision Systems (CVS). On the contrary, the visual dataflow-based environments are similar to the traditional schematic-based tools, which usually provide libraries composed of blocks with a high degree of functional abstraction that allow graphically constructing system models.

Simulink is an extension of the widely-used MATLAB environment that is specifically oriented for graphic prototyping and simulation of dynamical systems [7]. It also has a natural interface with MATLAB, so that its analysis and graphical representation tools can be used in the MATLAB workspace for post-processing and visualization. Like MATLAB, Simulink supports the extension of its functionalities by means of the add-in of application-specific libraries of components (*toolboxes*). Since the inclusion of toolboxes that allow the simulation and generation of hardware components, as the DSP Blockset toolbox or more recently the Xilinx System Generator (XSG) [5] and the Altera DSP Builder Altera [6] toolboxes, Simulink has become a powerful tool for HW/SW co-design [8, 9].

The Xilinx blockset contains high-level blocks that map intellectual property (IP) cores that have been handcrafted for efficient implementation in the target Xilinx FPGA. However, the XSG toolbox includes only some basic blocks that can be used as “bricks” for developing more complex structures. Based in these simple blocks, in this work the development of a visual processing library is presented. The library components have been optimized both in processing efficiency and in FPGA occupancy. Taking advantage of the XGS link with Matlab, the library blocks have been parameterized. This greatly eases the use of the library, as it can be used for a wide variety of applications without the need for the user of changing the code. Also, some Matlab-based software components have been implemented to allow co-simulation. They constitute the foundations of a complete HW/SW co-design and co-simulation Simulink-based scheme that will allow the rapid prototyping and implementation of hybrid CVS applications using Xilinx FPGAs.

The rest of this paper is organized as follows. The blocks of the visual processing library, jointly with a brief description of some of the underlying algorithms are explained in Section 2. The steps followed to design an HW/SW CVS using this library are briefly reviewed in Section 3. In Section 4, the use of this library to develop a complete study case is presented. Finally, conclusions and future research lines are included in Section 5.

## 2 The HW-SW Visual Processing Library

The designed CVS library consists of Simulink blocks. This library, whose purpose is the modelling and generation of HW-SW image processing and computer vision applications, behaves as any other Simulink library, and fully integrates with the Matlab/Simulink simulation environment. The design of a HW-SW system is thus performed by “dragging and dropping” the library blocks onto the Simulink editor, in which they are linked to construct the functional prototype.

### 2.1 HW Blocks

The HW components process the input pixels as they come in raster-scan order (from left to right and from top to bottom), so there is no necessity of having a whole image stored to begin the processing. This reduces the storage requirements and the amount of memory accesses, which generally constitute a bottleneck.

All the blocks have been homogeneously designed to assure interconnectivity. At each clock cycle, every block receives jointly with the input data (usually a pixel value) two control signals corresponding with the synchronization signals: line blank (LBL) and frame blank (FBL). Besides of the output data (generated at each cycle), the blocks include two additional output signals corresponding with the LBL<sub>o</sub> and FBL<sub>o</sub> control signals of the output stream.

The designed blocks are straightforwardly parameterized, so that they can be used for processing images with different sizes or formats without reprogramming. The arithmetic precision of the blocks in the data path is specified using Matlab expressions, making possible to minimize the hardware used, and avoiding the possibility of overflow. Therefore, changing parameters automatically gives an appropriately customized implementation.

Several image processing algorithms have been implemented using XSG.

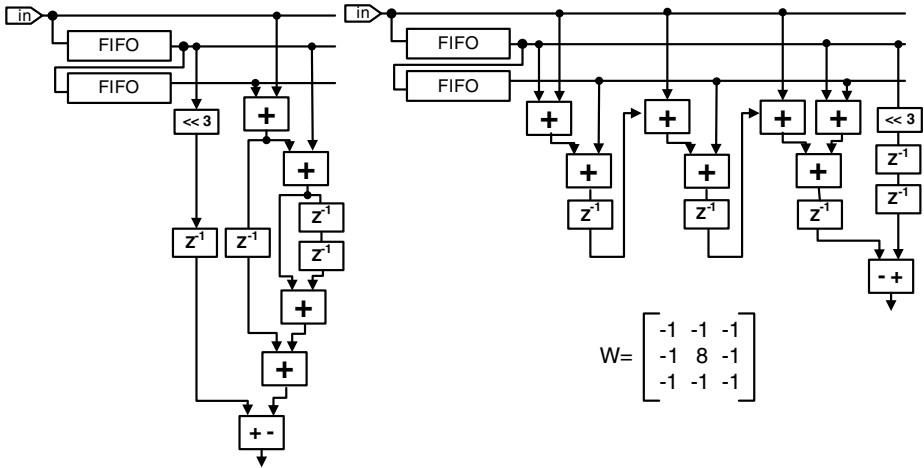
- Colour space conversions.
- Image cropping.
- Brightness/Contrast shift and scale.
- Threshold and double threshold.
- Specific filters: Gaussian, Laplacian, Prewitt, Sobel, Mean, Sharpening Median...
- Generic convolutions 3x3 and 5x5.
- Morphological binary operations with generic 3x3 and 5x5 structuring elements: erosion and dilation.
- Logical and Arithmetic operations: and, or, nor, add, sub...
- Connected Component Labelling.
- Calculus of the zero and first-order moments.

Some of them are detailed in the following lines.

Specific convolutions: As there exist some very frequently used convolution kernels, as the Prewitt, Laplacian and the Sobel kernels for edge detection, the Mean and Gaussian filters for noise filtering or the Sharpening kernel for image enhancement,

specific optimized code has been developed for them, to minimize resources while achieving high performance.

The input data stream arrives to the convolver in “row scan” format, which means that for every processing pixel it is necessary to wait until all the elements involved are available. Therefore, it is required a delay to store N-1 image lines (N being the row number of the convolution mask) that is implemented using N-1 FIFO memories.



**Fig. 1.** Two implementations of a 3x3 convolution. Left: proposed scheme. Right: Lisa’s scheme.

Lisa [10] offers a column-access architecture, which minimizes the amount of resources used in the FPGA by in-parallel processing the columns of pixels involved on each computation. Besides, by decomposing the mask weights in their binary representation, many multipliers can be replaced by shift-registers and adders. From the Lisa architecture, the convolutions have been optimized by decreasing the number of adders. To illustrate the followed procedure, in Fig. 1 the Lisa scheme for a simple filter is shown on the right, while the implemented one is shown on the left of the figure.

Another implemented algorithm that requires special mention is the connected component labelling. This algorithm is usually in the base of high-level image processing. Its input is a binary or grey-level image, while its output is a symbolic image, in which a label (usually a natural number) is attributed to each pixel in the image to symbolize that it belongs to an object represented by the label.

Since the shape of the object can be arbitrary, connected component labelling involves significant data computation and communication between the pixels in the image. To solve this problem, several sequential and parallel algorithms have been proposed [11]. In the library, the classical algorithm, which makes two forward raster scan passes through the image, has been implemented.

However, most times labelling is only required as a previous step for calculating properties of the objects as their masses, centres-of-mass or higher-order moments.



Taking this into account, special blocks have been constructed that perform these calculations without needing the second forward pass through the image, thus saving processing time and FPGA occupancy.

## 2.2 SW Blocks

Generic high-level image processing algorithms have been implemented by encapsulating in Simulink blocks some functions from the Image Processing Toolbox of MATLAB. In the construction of these wrappers, it has been taken into account that while the hardware processing is pixel-oriented, the software processing is frame-oriented. This causes synchronization problems, as the SW and HW will typically run on different frequencies. To tackle this, each SW block has been provided with an *enable* input that is marked as TRUE every time a frame is available from the HW.

For allowing the simulation of a whole system including acquisition, some blocks, which do not generate code in the final implementation, have been developed.

Camera blocks. Several camera blocks are provided, which model the behaviour of various common digitizers and non-interlaced digital cameras. Currently, there are available three kinds of blocks, giving YCrCb 4:2:2, Luminance and RGB output signals respectively. These blocks read one or more image files (they accept most of the common file formats for image storage) and provide, jointly with the corresponding pixel values, the appropriate LBL and FBL synchronization signals.

Viewer blocks. A number of viewer blocks have been built, to allow an easy inspection of the data flowing by the pipelines. At the moment, there are viewers for all the image types given by the camera blocks, plus several specific viewers that show the outputs of some blocks like the labelling block or the area and centre-of-mass blocks.

## 3 Design Flow

Using this library, to create a CVS application the user must only drag the corresponding blocks from the library and drop them into a Simulink empty model, then interconnect the blocks to form the application flow diagram. The constructed model can be simulated in Simulink, employing the stimuli and visualization blocks included in the library. This simulation allows verifying if the desired functionality has been achieved, and it is considerably faster than simulations performed by specific hardware simulators as ModelSim [12]. Due to the library HW blocks are made up of XSG simple blocks, for the HW the simulation results are identical to those that would be obtained in the real FPGA implementation.

After simulation, if the functional requirements are met, the user must decide on the target platform for the HW partition. Once the target platform has been selected, the hardware code (VHDL) can be automatically generated. This code incorporates optimized Xilinx LogiCORES, thus assuring that the implementation will be efficient.

The generated HW code is automatically encapsulated inside a VHDL project, in which the specific code for I/O interfacing related to the selected platform is in-

cluded. This is required because XSG is a good setting in which to implement data paths, but is less well suited for sophisticated external interfaces that have strict timing requirements (for example, it can not work with several external clock sources). To tackle this, a HDL wrapper has been created that automatically generates the necessary code for the I/O interfaces, i.e. the video transfer from the digitizer (or from the camera), the transfers to/from external memories and the data transfer to/from SW blocks. As said before, this wrapper is specific for each HW target platform.

This enveloping VHDL project can then be automatically synthesized. As a result of the synthesis, the cost (measured in FPGA area occupancy), the maximum working frequency for the FPGA and the HW execution time are obtained.

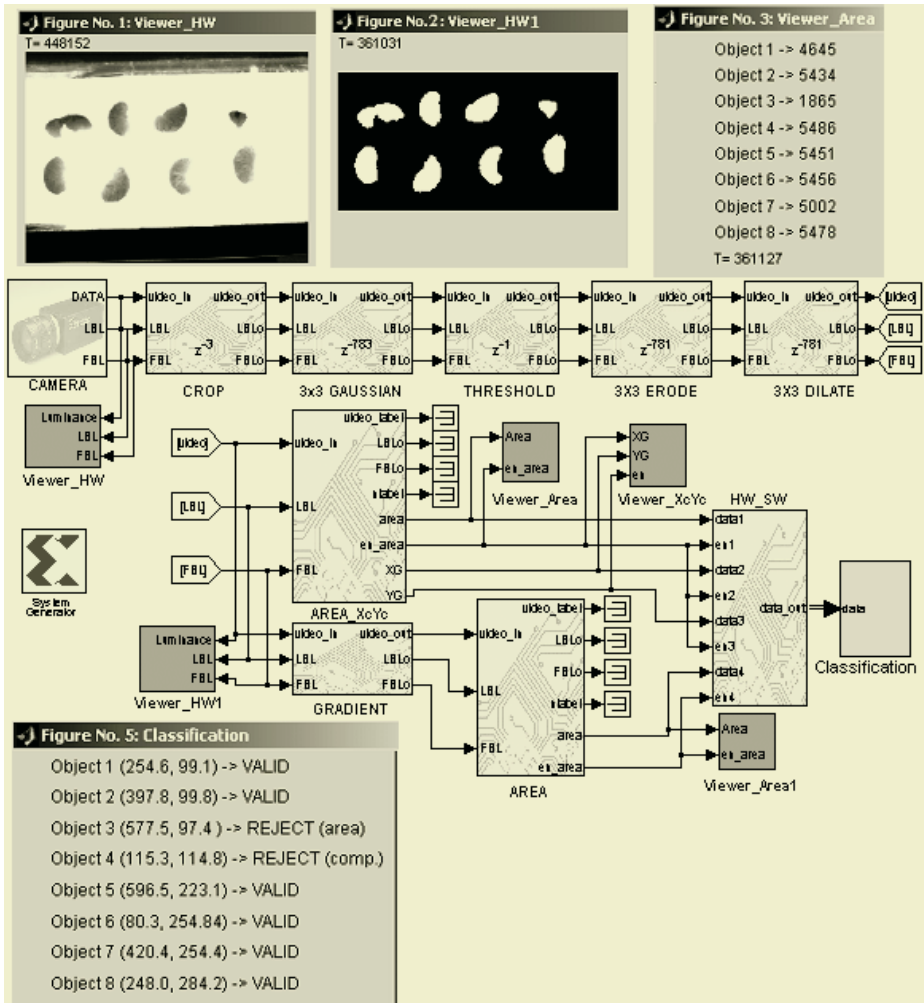
Finally, if the temporal requirements are fulfilled, the FPGA on the HW platform is configured with the bitstream file (automatically generated during the synthesis), by means of the appropriate tool provided by the manufacturer.

## 4 A Practical Study Case

In order to test the capabilities of the library, a CVS for tangerine segment inspection has been constructed from scratch. The objective of the CVS is to reject the tangerine segments that appear split in pieces, or that are simply too small to be canned. This must be done in real-time, because the inspection system is working on line in the final step of the canning line. This canning line is composed of a conveyor belt, which transports the tangerine segments under a camera and through several air-jet ejectors to the canning mechanism.

The scheme of the HW/SW proposed CVS is shown in Fig 2. The algorithm works as follows: as the pixel information is submitted from the camera, a cropping is performed to discard pixels from areas outside the conveyor belt. On the passing pixels, a 3x3 Gaussian filter (with standard deviation = 0.9) is applied to remove noise. Thus, a thresholding is carried out to separate the tangerine segments (dark) of the conveyor belt (bright). A binary opening (erosion + dilatation) is then performed with a 3x3 solid structural element, to remove binary noise in the form of small blobs. After this, the pipeline forks. In one of the lines, the binary image is processed to obtain the area and center-of-mass of the blobs. This information is fed into a HW/SW interface block which pumps the data into the SW blocks. Simultaneously, in the other line, edge detection is carried out, so that a binary image containing the borders of the blobs is obtained. This image is processed by a block that gives the area of these borders. In this way, a measure of the perimeter of the blobs is obtained and, though the HW/SW interface block, it is fed to the SW processing blocks.

The HW part of the algorithm was implemented in a Nallatech Ballynuey 3 card [13]. This card is a general-purpose PCI board, specially designed for prototype development, and it is based on a Virtex 2V3000 FPGA. It incorporates two external ZBT SSRAM memories (8 Mb) and four DIME slots of expansion. In one of them, a Nallatech Ballyvision module was connected, to allow input from an external PAL camera.



**Fig. 2.** Simulink co-simulation screenshot. SW blocks are shown in a plain-color while patterned ones denote HW components. Images resulting from some SW/HW processing steps are shown together with the corresponding temporization cycles.

As a result of the synthesis on the Virtex 2V3000 FPGA, the proposed hardware architecture is able of processing 778x576 images with a cycle time of 11.62 ns (82 MHz). This implies that more than 190 frames per second can be processed, thus allowing a high pace to the conveyor belt.

## 5 Conclusions

In this work, a CVS library has been developed, which allows using the Matlab/Simulink environment for prototyping, co-simulating and automatic HW code

generation of HW/SW computer vision systems. The hardware blocks, based on the XSG tool, have been parameterized and optimized. Also, the HW code generation has been fully automated, including wrapping mechanisms that extend the original capabilities of the XSG. The result is a library of blocks fully integrated in Matlab/Simulink that greatly eases the functional prototyping, verification and final implementation of HW/SW computer vision systems without the necessity of mastering neither a hardware description language nor the intricacies of the hardware platform.

To test the CVS library, a HW/SW computer vision system for the automated inspection of tangerine segments has been constructed from scratch, with excellent results.

## References

1. Arnout G., C for System Level Design, Proceedings of Design, Automation and Test in Europe Conference and Exhibition, March 1999.
2. Panda P.R., SystemC - a modeling platform supporting multiple design abstractions, Proceedings of the 14th ISSS, 2001 pp. 75 -80
3. Hwang J, Milne B, Shirazi N, Stroomer J., System Level Tools for DSP in FPGAs. FPL 2001, Lecture Notes in Computer Science, pp 534-543.
4. SystemC. Available: <http://www.systemc.org>
5. System Generator: Reference guide, <http://www.xilinx.com/>
6. DSP builder. Available: <http://www.altera.com/>
7. The Math Works Inc., <http://www.mathworks.com>
8. Líčko M., Schier J., Tichý M., Kühn M.: MATLAB/Simulink Based Methodology for Rapid-FPGA-Prototyping. FPL 2003.Vol. 2778, pp 984-987.
9. Denning D., Harold N., Devlin M., Irvine J.: Using System Generator to Design a Reconfigurable Video Encryption System. In: P.Y.K. Cheung et al. (Eds.): FPL 2003. Lecture Notes in Computer Science, Vol. 2778, pp 980-983.
10. Lisa F, Cuadrado F, Rexachs D, Carrabina J: A reconfigurable coprocessor for a PCI-based real time computer vision system. FPL 1997. pp 392-399.
11. Wang K., Chia T., Chen Z., Lou D.: Parallel Execution of a Connected Component Labeling Operation on a Linear Array Architecture. Journal of Information Science and Engineering 19, pp 353-370 (2003).
12. ModelSim. Available: <http://www.model.com>
13. Nallatech. <http://www.nallatech.com>

# 2-D Discrete Cosine Transform (DCT) on Meshes with Hierarchical Control Modes

Cheong-Ghil Kim, Su-Jin Lee, and Shin-Dug Kim

Supercomputing Lab, Dept. of Computer Science, Yonsei University  
134 Shinchon-Dong, Seodaemun-ku, Seoul, Korea 120-749  
{cgkim, heagy, sdkim}@parallel.yonsei.ac.kr

**Abstract.** An effective matrix operation is critical to process 2-D DCT. This paper presents a hierarchically controlled SIMD array (HCSA) well suited to matrix computations, in which a conventional 2-D torus is enhanced with the hierarchical organization of control units and the global data buses running across the rows and columns. The distinguished features of the HCSA are the diagonally indexed concurrent broadcast and the efficient data exchanges among PEs through either row or column broadcast. Therefore, the HCSA can provide significant improvement on computation steps of DCT. For the performance evaluation, an algorithmic mapping method is used and the number of computation steps is analytically compared with semisystolic architecture.

## 1 Introduction

The demand of high-speed computing architecture for discrete cosine transform (DCT) has been increased continuously due to the dominant popularity of digital signal processing and video compression. Moreover, it has been included in current image/video standard specifications.

The primitive operation of DCT is based on matrix computations in which parallel processing techniques must be considered for real time processing with large 2-D data-sets. This operation is characterized as data intensive tasks accompanied by heavy memory accesses; on the other hand, their computational complexities are relatively low. Thus, it naturally maps onto SIMD (Single Instruction Multiple Data stream) parallel processing on 2-D array processors with distributed memory. Corresponding to the growing fabrication technology and CAD tool advances, which allow the implementation of a complex system on a chip, SIMD arrays become increasingly important as coprocessors in domain specific systems with the form of application specific integrated circuits (ASICs) [1, 2] and reconfigurable systems [3].

This research proposes a modified 2-D SIMD array architecture, called hierarchically controlled SIMD array (HCSA), in which a conventional 2-D mesh of  $n^2$  processors with wrap-around links is enhanced with the hierarchical organization of control units and the global data buses. They enable efficient data movements on the proposed architecture, so that the HCSA is well suited for implementing 2-D DCT using row-column decomposition method by reducing the computation steps and cycles.

In the next section, we describe the DCT algorithm and the computational complexity. Section 3 introduces the architecture and operational model of the HCSA system. In Section 4, 2-D DCT is mapped on the HCSA and the result is compared in Section 5. Finally, the paper ends with conclusions in Section 6.

## 2 2-D DCT Algorithm

The 2-D DCT can be described as a transformation from a 2-D matrix of pixels to that of spatial frequency information. The transformed matrix contains many small values or zero entries, so that the compression of such data using standard techniques becomes very straightforward.

For an input matrix  $x(m, n)$  and an output matrix  $z(k, l)$  with  $\{0 \leq m, n, k, l \leq N-1\}$ , the forward  $N \times N$  2-D DCT is defined as

$$z(k, l) = \frac{2}{N} \alpha(k) \alpha(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) \cos \frac{(2m+1)\pi k}{2N} \cos \frac{(2n+1)\pi l}{2N} \quad (1)$$

where

$$\alpha(k) = \alpha(l) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } k = l = 0 \\ 1, & \text{otherwise} \end{cases}$$

Equation 1 can be rewritten in matrix form as

$$Z = AXA^T, \quad (2)$$

where  $X$  is the source pixel (spatial domain) data,  $Z$  is the DCT output coefficients (frequency domain), and  $A$  is an orthogonal matrix defined as

$$a(u, v) = \sqrt{\frac{2}{N}} \alpha(u) \cos \frac{(2v+1)\pi u}{2N} \quad (3)$$

A naive implementation of Equation 1 requires  $N^4$  multiplications. Alternatively, the 2-D DCT can be computed by applying the 1-D DCT by rows (columns) and then, by columns (rows) due to separable property of 2-D DCT. This approach is called the row-column decomposition method and requires  $2N$  instances of  $N$ -points 1-D DCT to implement an  $N \times N$  2-D DCT, resulting in  $2N^3$  multiplications.

The 1-D DCT may be computed by using a direct approach based on the fast cosine transform (FCT) method [4] or an indirect strategy based on the discrete Fourier transform [5]. Even though the indirect approach usually requires more number of operations than the direct approach, it has been shown [6] that efficient algorithms can be obtained by transforming two input data streams simultaneously. These approaches need to transpose the intermediate results by using a memory array, thus leading to a high circuit complexity and a long latency for loading and unloading.

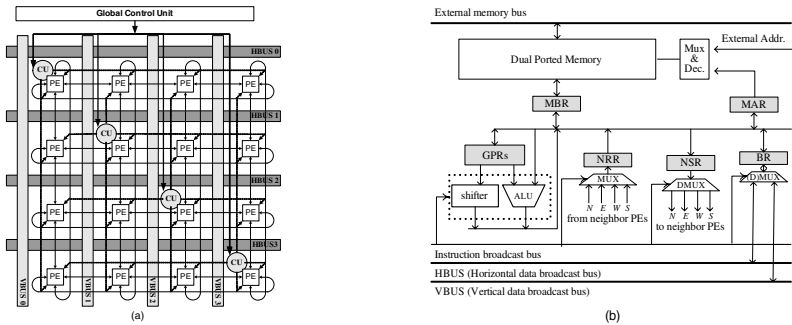
Another approach to compute directly the 2-D DCT was proposed without decomposing it into two successive 1-D DCTs [7]. Although this approach requires the least number of multipliers and adders, the structure of the resulting architecture is very complicated and the interconnection complexity is high.

### 3 The Architecture and Operational Model

The HCSA consists of a global control unit (GCU),  $n$  local control units (CUs),  $n^2$  processing elements (PEs), and  $n$  row buses and  $n$  column buses, as shown in Fig. 1(a). All PEs are connected with the torus interconnection. Both CUs and PEs have their own memory, called the control memory (CM) and the processing memory (PM), respectively. Here, an example of  $4 \times 4$  PE configuration is shown.

Differently from the conventional SIMD array having a single control unit, the control system of the HCSA is configured hierarchically, consisting of a GCU and  $n$  local CUs. The GCU controls and synchronizes all  $n$  local CUs, providing all the interfaces with outsides for programming and communications. The  $n$  local CUs are configured diagonally and controls the PEs connected to its corresponding group. In addition, two types of global bus are enhanced for horizontal data broadcast with row-control mode (RC-mode) and vertical data broadcast with column-control mode (CC-mode).

The HCSA could be utilized for the matrix-oriented data intensive applications with efficient data movement. First, the HCSA can allow the diagonally indexed concurrent broadcast, which enables the efficient delivery of each operand vector to other PEs at the same time rather than sending each element of an operand vector that is a common way in conventional 2-D SIMD arrays. Therefore, the matrix-by-vector products can be performed in a single cycle because the transmission operation of an operand vector can be overlapped with multiplication. Second, the HCSA can provide a direct connection between the PEs connected to the same bus by the RC-mode and CC-mode, which are especially suitable for the computational model of DCT.



**Fig. 1.** Architecture: (a) Overall structure, (b) PE organization.

Each PE in the HCSA is a simple processor consisting of an ALU, a shifter, a set of general-purpose registers, several special-purpose registers, and dual ported memory as shown in Fig. 1(b). Special purpose registers are devised for row/column broadcasting and neighboring PE communications. Under SIMD mode, all PEs are receiving instructions from local CUs and executing same operation with different data.

As described above, the HCSA system has two different types of controllers: the GCU and local CUs. The GCU performs the interaction between the HCSA and the outer system, i.e., host processor (HP) or master processor (MP). The interaction mechanism of the control transfer between the outer system and HCSA is performed via the conventional subroutine calling mechanism, which can initiate its corresponding HCSA subroutine call in order to differentiate it from any other conventional subroutine calls. The local CUs directly perform the control of their associated PE groups.

The control mode of the HCSA system is classified as RC-mode and CC-mode. In the RC-mode, each local CU can control the same row PE group and in the CC-mode, the same operation is performed on the column basis. The overall execution flow is as follows. First, the HP compiles an application program and stores it on the secondary storage. When this program is to be executed, the HP loads that program into its memory. At this moment, the parallel code and data blocks are loaded into the CM and the PMs, respectively. After that the HP starts executing that program. When the HP encounters any calling instruction to initiate the HCSA, the control is transferred to the GCU of HCSA. And this control will be returned back to the HP when the GCU completes the called HCSA subroutine. Once the HCSA is initiated, the GCU sequentially broadcasts the parallel instructions to the local CUs, and then local CUs broadcast the control signals to their associated PE groups concurrently.

## 4 2-D DCT on HCSA

The HCSA system can effectively map 2-D DCT using the row-column decomposition approach without the transposed matrix that is generally required in the existing architectures [1]. For our mapping processes, the following assumptions for the HCSA system and 2-D DCT are made. The source input matrix  $X$ , the DCT coefficient matrix  $A$ , and its transposed matrix  $A^T$  for 2-D DCT are supposed to be stored in the memory block of each PE prior to the processing. The size of each matrix and the number of DCT block number of any source input are assumed to be  $N \times N$  and  $L$ , respectively. The size of macro block is  $8 \times 8$ , so that it is assumed that the number of processing units, i.e.,  $P$ , is not smaller than  $8^2$ .

The pseudo code of the proposed 2-D DCT algorithm is shown in Fig. 2(a). Here, each  $PE_{i,j}$  for all  $0 \leq i, j \leq \sqrt{P}-1$  computes in parallel by accessing the memory block  $PM_{i,j}$ . First, every  $PE_{i,j}$  fetches the element  $a_{i,j}$  of  $A_{i,j}$  and  $a_{j,i}$  of  $A_{j,i}$  into their local registers in parallel in line 3 and 4. Second, every  $PE_{i,j}$  fetches the element  $x_{i,j}[l]$  of the  $l$ -th DCT block matrix  $X[l]$  to their local registers in parallel in line 8. Third, the parallel 1-D row DCT procedure,  $rowDCT-HCSA(l)$  and the parallel 1-D column DCT procedure,  $colDCT-HCSA(l)$ , are consecutively called shown in Figs. 2(b) and 2(c). Finally, every  $PE_{i,j}$  stores the element  $z_{i,j}$  of the DCT output matrix  $Z_{i,j}[l]$  to their local memory. Those 2~4 Steps are iteratively performed for  $l = 0, 1, \dots, L-1$  as shown in lines 6~15 of Fig. 2(a).



The row DCT procedure, *rowDCT-HCSA(l)* is shown in Fig. 2(b) and the computing process is described as follows. First, every  $PE_{m(j),j}$  ( $m(j) = (j+k) \bmod \sqrt{P}$ ) concurrently broadcasts the element  $a_{j,m(j)}$  to its own column processor group by CC-mode; and then every  $PE_{i,j}$  multiplies the operand broadcasted by the element  $x_{i,m(j)}[l]$  and accumulates the result product as shown in lines 5~7. After finishing the multiply-and-accumulate (MAC) operation, every  $PE_{i,j}$  shifts the element  $x_{i,m(j)}[l]$  to the left as shown in line 8. The above processes are repeated until finishing the computation for the entire matrices. The computing process of the column DCT procedure, *colDCT-HCSA(l)* is basically the same as the row DCT procedure, except broadcasting mode and shift direction. The column DCT procedure uses RC-mode for broadcasting and operand shift up shown in Fig. 2(c).

```

Algorithm 1. 2-D DCT
{ 2-D DCT procedure on the HCSA }

1. procedure 2D-DCT ( )
   // 2-D DCT (row-column decomposition method)
2. parbegin all 0 = i = N-1, 0 = j = N-1 do { DCT coefficients loading }
3.   PEi,j reads ai,j;           { for rowDCT-HCSA }
4.   PEi,j reads ai,j;           { for colDCT-HCSA }
5. parend {line 2 parbegin}

6. for l = 0 to L-1 do
7.   parbegin all 0 = i = N-1, 0 = j = N-1 do { input pixel loading }
8.     PEi,j[l] reads xi,j[l];
9.   parend {line 7 parbegin}
10.  rowDCT-HCSA(l); // call procedure rowDCT-HCSA
11.  colDCT-HCSA(l); // call procedure colDCT-HCSA
12.  parbegin all 0 = i = N-1, 0 = j = N-1 do { output pixel storing }
13.    PEi,j[l] writes zi,j[l];
14.  parend {line 12 parbegin}
15. endfor {line 6 for}

16. endprocedure (a)
    
```

```

Algorithm 2. Row DCT Stage on HCSA
{ define macro for index }
1. #define m(x) ((x+k) mod N)
{ Row DCT procedure on the HCSA }
2. procedure rowDCT-HCSA ( l ) { Y = XAT }
3. for k = 0 to N-1 do
{input-by-transposed DCT coefficient MAC operation }
4.   parbegin all 0 = i = N-1, 0 = j = N-1 do
5.     PEm(j),j[l] broadcasts aj,m(j) by CC-mode;
6.     PEi,j[l] computes xi,m(j)[l] * aj,m(j);
7.     PEi,j[l] computes yi,j[l] += xi,m(j)[l] * aj,m(j);
8.     PEi,j[l] shifts left xi,m(j)[l];
9.   parend {line 4 parbegin}
10.  endfor {line 3 for}
11. endprocedure
    
```

(b)

```

Algorithm 3. Column DCT Stage on HCSA
{ define macro for index }
1. #define m(x) ((x+k) mod N)
{ Column DCT procedure on the HCSA }
2. procedure colDCT-HCSA ( l ) { Z = AY }
3. for k = 0 to N-1 do
{DCT coefficient-by-rowDCT value MAC operation }
4.   parbegin all 0 = i = N-1, 0 = j = N-1 do
5.     PEi,m(j)[l] broadcasts ai,m(j) by RC-mode;
6.     PEi,j[l] computes ai,m(j) * ym(j),i[l];
7.     PEi,j[l] computes zi,j[l] += ai,m(j) * ym(j),i[l];
8.     PEi,j[l] shifts up ym(j),i[l];
9.   parend {line 4 parbegin}
10.  endfor {line 3 for}
11. endprocedure
    
```

(c)

**Fig. 2.** Algorithms: (a) 2-D DCT, (b) Row-DCT, (c) Column-DCT.

Consequently, the number of computation steps required for the parallel 1-D row DCT and column DCT by the HCSA system can be obtained as

$$\sum_{n=0}^{N-1} \{ T_{broadcast} + T_{mult} + T_{add} + T_{send} \} = 4N \quad (4)$$

In the above equation, each of  $T_{broadcast}$ ,  $T_{mult}$ ,  $T_{add}$ ,  $T_{send}$ , and  $T_{store}$  is assumed to take one unit time. According to Algorithm 1 of Fig. 2(a) and Equation 4, the number of computation steps required to perform the DCT of the  $L$  input block by the HCSA system can be obtained as

$$\overline{2}^{memory} + \sum_{l=0}^{L-1} \left\{ \hat{1}^{memory} + \overline{4N}^{row\ DCT} + \overline{4N}^{column\ DCT} \right\} = 2 + L(8N + 1) = 8LN + L + 2 = O(LN) \quad (5)$$

Therefore, the total number of cycles for computing 2-D DCT becomes  $2LN$  and a  $4 \times 4$  2-D DCT requires  $8 (= 2 \times L \times N = 2 \times 1 \times 4)$  cycles. The communication steps of the row DCT,  $Y = X \times A^T$  and the column DCT,  $Z = A \times Y$  are illustrated in Figs. 3(a) and 3(b) for  $N=4$  and  $P=16$ , respectively.

As a result, the HCSA system does not require the transpose memory for transposition, which consumes much area for global connections and much time for loading and unloading. In addition, the HCSA system is not restricted by the transform length  $N$  to be a prime number [8] or an integer power of 2 because this proposed system can be easily scaled without modifying the basic control scheme and PE structure.

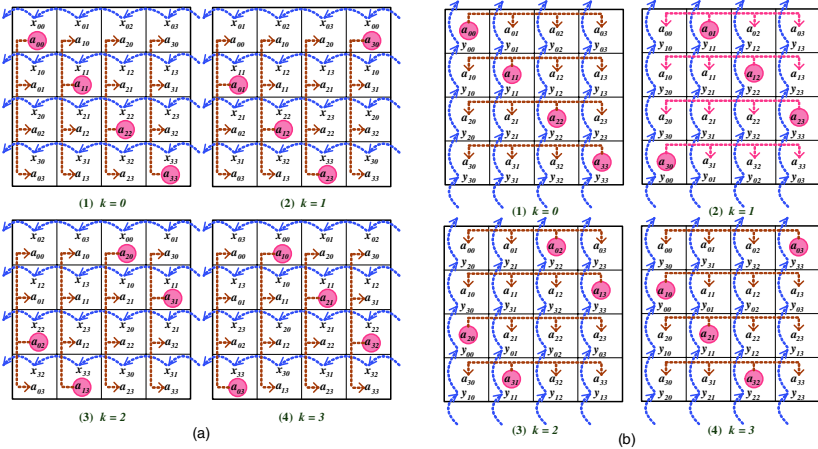
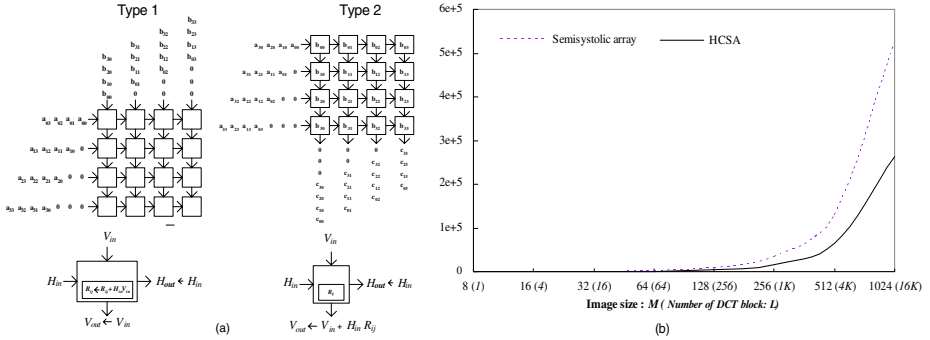


Fig. 3. Communication steps: (a) Row DCT  $Y = X \times A^T$ , (b) Column DCT  $Z = A \times Y$ .

### 5 Performance Evaluation

Lim *et al.* proposed semisystolic arrays for the unified computation of 2-D DCT that did not require transposition [8], in which an intermediate approach was allowed to design an architecture having throughput and circuit complexity between the extremes of the previous cases. Therefore, this semisystolic architecture is compared with the proposed algorithm onto HCSA in terms of computation cycles.

Kung [9] proposed two types of semisystolic array for the multiplication of two  $N \times N$  matrices: Type 1 and Type 2. In Type 1, output data are not produced in the boundary cells of the array; in Type 2, input data are needed to be preloaded into the cells of the array. Fig. 4(a) depicts the architecture of two types of semisystolic array for matrix multiplication  $C=AB$  with  $N=4$  with their PE structure. Here,  $H_{in}$  and  $H_{out}$  represent the horizontal input and output, respectively;  $V_{in}$  and  $V_{out}$  represent the vertical input and output, respectively.  $R_{ij}$  is a value saved in a register of the the  $ij$ -th PE.



**Fig. 4.** Performance comparison: (a) Semisystolic array, (b) The computation cycles as the image size  $M$  changes ( $P=64$ ).

In Type 1 semisystolic array shown in Fig. 4(a), each PE performs a multiply-accumulate operation. In  $N$  cycles, each PE computes an inner product of a row of the horizontal input and a column of the vertical input. The latency is defined as the elapsed time between the first data entry point and the moment when output data are available. The cycles per datum (CPD) is the number of clock cycles to compute each point of the transform which is an indication of the average latency. This  $N \times N$  systolic array has a latency of  $3N-2$  and a CPD of  $N$ . This array can be defined as semisystolic since the output data are not produced in the boundary cells of the array, such that it has overhead for the output to be shifted out of the array. In Type 2 semisystolic array, to compute  $C=AB$ , each component of matrix  $B$  is preloaded into the array with one element of the matrix in a register within each PE, while matrix  $A$  is fed into the array. Each PE multiplies the horizontal input by the register value and adds this to the vertical input to produce the vertical output. The inner product of a column of the input matrix and a column of the stored matrix is computed for every  $N$ . In [8], two types of systolic arrays are combined into one array, so that input and output move along the axes and intermediate result does not move. In this way, the systolic array does not require any transposition and the total cycles for computing 2-D DCT becomes  $(4N-2)L$  when  $L$  is the number of DCT blocks.

Consequently, the number of computation cycles on the HCSA with its corresponding algorithms could be reduced significantly compared with that of the semisystolic array. Furthermore, the semisystolic array requires the different behaviors of PEs because of combining the two types of semisystolic arrays; on the other hand, HCSA performs the same behaviors of PEs owing to using the SIMD execution.

For performance evaluation, an  $8 \times 8$  DCT block ( $N=8$ ) is used. Fig. 4(b) shows the number of computation cycles as the size of an image varies from  $8 \times 8$  to  $1024 \times 1024$  when the number of processors is 64. The value in parenthesis of the  $x$ -axis presents the number of DCT blocks in the corresponding image size. According to Fig. 4(b), the performance is improved in proportion to the size of image, i.e., the number of DCT blocks, and also it shows that the performance of HCSA is better than semisystolic array by reducing 46.7%~50.0% of the total cycles.

## 6 Conclusion

A new 2-D torus is presented. It is enhanced with the hierarchical organization of control units and the global data buses and targeted at achieving a high performance on the matrix computations of 2-D DCT. It has the advantages of the diagonally indexed concurrent broadcast and the efficient data movements between PEs. Therefore, the proposed array can achieve considerable performance gains on matrix computations by reducing data movements frequently occurred in previous SIMD architectures, so that it is suited to the operations for matrix-by-vector and matrix-by-matrix multiplication. For the performance evaluation, an algorithm mapping method is used and the computation step is compared analytically with the semisystolic array. The HCSA could reduce 46.7% ~ 50.0% of the cycles required by 2-D DCT algorithm mapped on the semisystolic array.

## References

1. Smith, R., Fant, K., Parker, D., Stephani, R., Ching-Yi, W.: An Asynchronous 2-D Discret Cosine Transform Chip. In Proc. Int'l Symp. Asynchronous Circuits and Systems (1998) 224-233
2. Cho, N.I., Lee, S.U.: DCT Algorithms for VLSI Parallel Implementation. IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. 38. (1990) 121-127
3. Bagherzadeh, N., Filho, C., Lu G., Kurdahi F.J., Lee M.-H., Singh, H.: MorphoSys: an Integrated Reconfigurable System for Data-parallel and Computation-intensive Applications, IEEE Trans. Computers, Vol. 49, Issue: 5. (2000) 465-481
4. Sheu, M., Lee, J., Wang, J., Suen, A., Liu, L.: A High Throughput-rate Architecture for 8×8 2D DCT. In Proc. Int'l Symp. Circuits and Systems Vol. 3 (1993) 1,587-1,590,
5. Makhoul, J.: A Fast Cosine Transform in One and Two Dimensions. IEEE Trans. Acoustics, Speech, and Signal Processing Vol. 28. (1980) 27-34
6. Vetterli, M., Nussbaumer, H.J.: Simple FFT and DCT Algorithms with Reduced Number of Operations. Signal Processing Vol. 6. (1984) 267-278
7. Cho, N., Lee, S.: Fast Algorithm and Implementation of 2D Discrete Cosine Transform. IEEE Trans. Circuits and Systems Vol. 38. (1991) 297-305
8. Lim, H.S., Piuri, V., Swartzlander Jr., E.E.: A Serial-Parallel Architecture for Two-Dimensional Discrete Cosine and Inverse Discrete Cosine Transforms. IEEE Trans. Computer, Vol. 49. (2000) 1,297-1,309
9. S.Y. Kung. VLSI Array Processor. Printice Hall, Englewood Cliffs, N.J. (1988)

# Domain-Specific Codesign for Automated Visual Inspection Systems

Sergio Cuenca<sup>1</sup>, Antonio Cámara<sup>1</sup>, Juan Suardíaz<sup>2</sup>, and Ana Toledo<sup>2</sup>

<sup>1</sup>Departamento de Tecnología Informática y Computación, Universidad de Alicante  
Alicante, Spain  
sergio@dtic.ua.es

<sup>2</sup>Departamento de Sistemas y Electrónica, Universidad Politécnica de Cartagena  
Murcia, Spain  
{Juan.Suardiaz, Ana.toledo}@upct.es

**Abstract.** In this paper we present a codesign methodology for high-performance Automated Visual Inspection systems (AVIs). The proposal consists in reference hardware/software architecture and its associated co-verification environment. The codesign method is stepwise refinement-based process that starts with a preliminary hw/sw partition based on the reference architecture. During refinement the selected hardware blocks are coded using the high level language Handel-C, and the rest of the system using a *plugin* library. This library allows to model different external components to hardware (software, external devices, etc...) with a behavioural, timing and performance view using software languages like C/C++. As a result of this design flow, we are able to verify and develop AVI systems with a significant improvement on traditional hardware/software codesign times.

## 1 Introduction

Product inspection is an important part of today's highly competitive industrial production (textile, agriculture and food, canning, etc..). The implementation of automatic systems that carry out this tasks show multiple advantages, however factors as throughput or economic cost are decisive in order to evaluate its viability. For applications that have to be integrated in production lines (on-line AVIs), which generally require a high speed response, hardware/software systems based on reconfigurable devices (mainly FPGAs) are, a priori, a valid alternative compared with commercial systems based on general purpose microprocessors and/or DSPs [1] [2]. Indeed, reconfigurable devices are an easy way to implement specific pieces of hardware for accelerating the more time consuming processes of a system. However, due to the processing heterogeneity that inspection algorithms perform, FPGAs are, at the moment, unable of supporting entirely this application specific domain and it is necessary a combined use of hardware and software components to reach both economics and performance requirements. Nevertheless, although has been reported some reconfigurable prototypes developed for this aim [3], the number of them implanted on industrial environments is very limited. One of the main reasons is that programming, verifying and fine tuning of these hardware/software systems is still a costly task that increases the cost and the development time.

In this respect, several tools have been developed for the simulation of image processing systems coded with HDLs (hardware description languages) [4], [5]. Due to the low-level of the description style (normally structural or RTL) we get very realistic performance and area estimations but this generally results in very long simulation times, and is thus rarely used. In addition, HDLs are not very popular between the vision systems developers that normally work with software high-level languages. Other approaches are being adopted from system level view using languages that support both hardware and software specification, e.g. SystemC [6] or MATLAB/Simulink + Xilinx System Generator [7]. In this way, the functional co-verification can be performed quickly but, at this level, we get very limited information about performance or resource usage. Therefore, a later HDL low-level simulation is needed.

Our proposal is a hybrid approach that makes use of Handel-C, a high-level HDL that support the description of hardware with the syntaxes of C/C++. The co-verification is supported on commercial Celoxica DK (a toolset around Handel-C) [8], which provides cycle-accurate simulation and gate count so throughput and resource usage estimations can be performed at the same time that functional specifications are verified. As start point for the codesign we propose the *onAVI* reference architecture for the specific domain of *on-line* AVIs, that comprises the functionalities of a wide range of these systems, and of which several approximations have already been implemented with good results [9]. It serves as preliminary hw/sw partition for successive refinements, limiting the number of decisions the designer has to take and, thereby providing faster development time through extensive design reuse.

This paper is organized as follows. In Section 2, the features of a generic architecture for AVIs and a preliminary partition are proposed. In Section 3, the codesign environment and codesign flow are described. Finally, some experimental results are shown in Section 4 and conclusions are given in Section 5.

## 2 *onAVI* Reference Architecture for *On-Line* AVI Systems

In general, although there is confirmation of multiple commercial inspection systems implanted, in no case exists a methodology that permits, regardless of the characteristics of the process, to entirely configure the AVI system. For this reason, the works developed show, mainly, particular solutions attending the inspection conditions. Even so, reviewing scientific literature on the matter and basing on our experience, it has been possible to detect some common subsystems.

The following subsystems form our proposal for the generic architecture (fig. 1): *Image acquisition subsystem*: it is in charge of illumination control, capture, conditioning and images storage. *Electro-Mechanical subsystem*: it carries out two tasks mainly; on one hand it prepares products to be inspected (alignment, positioning, etc) sending signals to notify when the product is ready to be inspected, on the other hand it receives signals from the control subsystem to handle the product once it is inspected (defects labelling, quality grading, products rejection, etc...). *Processing subsystem*: it run vision and classification algorithms. *Control subsystem*: it controls the whole system, it receives signals from the mechanic subsystem and decides when must be triggered the camera. It also receives the inspection results and decides the

actions that are going to be carried out to inspected products. *Management subsystem*: it includes the user interface. It manages the inspection results and allows the user to start and stop the system as well as configure the inspection parameters.

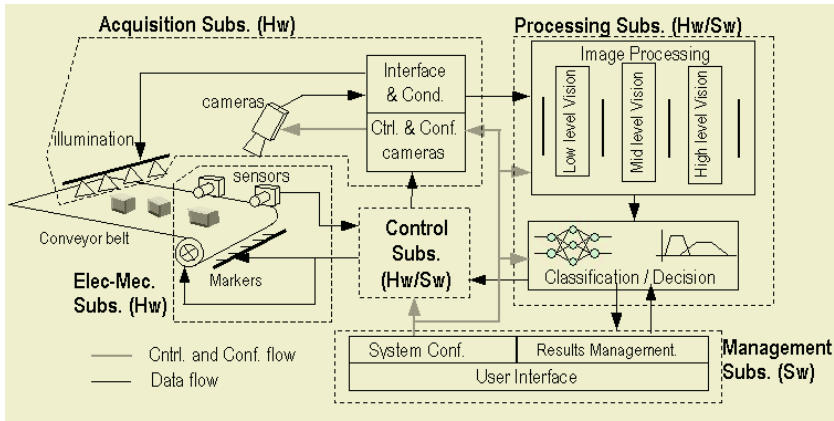


Fig. 1. Subsystems and preliminary Hw/Sw partition

From the study of the nature of the different tasks that perform these subsystems, two different computation models can be observed. In first place, a fine grain computation with high level of parallelism. This computation involves a large amount of data (integer type) on which it is carried out simple and repetitive operations (fixed point arithmetic and logic). This type is characteristic of tasks associated with data acquisition and low/middle level vision algorithms.

In second place, a coarse grain computation, that is carried out on a smaller amount of data (real type or integer type) but with multiple complex operations (fixed point and/or floating point arithmetic). This type of computation comprises high level vision algorithms, classification algorithms, management and control. These two computational models define a preliminary Hw/Sw partition (fig. 1) that serve as start point to successive refinement.

### Architecture Description

We propose the use of a reconfigurable computation system to implement the first type of processes, attached to a general purpose microprocessor to run the second. The main blocks of *onAVI* are outlined in Figure 2. The interface between hw and sw was implemented with two blocks. The hardware block, *SwIfware*, is in charge of the communication with the host for single word transfers (command, status and parameters) and DMA transfers. The drivers and communication subroutines that run on the microprocessor compose the software block, *HwIfware*. The PCI bus was used to communicate both sides of the architecture, in order to guarantee the necessary bandwidth. In addition, a double buffer (Bank0 and Bank1 in the figure) is included to allow hw and sw blocks to work concurrently. to which both hw and sw can access exclusively in a “ping-pong” fashion. This double buffer can be a shared resource or a local resource of the hw.

In order to achieve *on-line* processing, the work cycle is determined by the time of capturing a frame, a frame can be composed by several images, ( $T_F$ ), which includes the time between the capture of consecutive images ( $T_{IB}$ ) and the capture time for each image ( $T_I$ ). Another important issue to consider is the time of data transfer between Hw and Sw ( $T_T$ ), since it limits the maximum time that the sw has to complete its tasks.

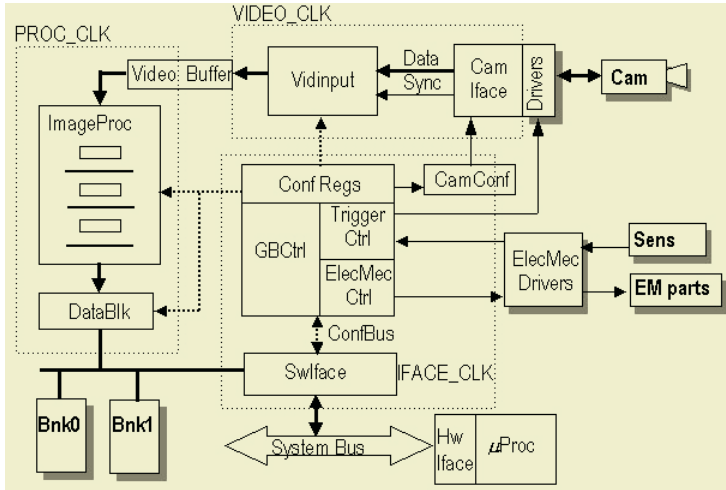


Fig. 2. Overview of *onAVI* reference architecture

With this approach, the *on-line* processing requirements are reduced to the  $T_{FPH}$  and  $T_{FPS}$  constrains:

$$\text{Max. time for frame processing in Hw: } T_{FPH} \leq T_F = n_i * (T_I + T_{IB}); \quad n_i \equiv \text{images/frame} \quad (1)$$

$$T_I = n_l * (T_L + T_{LB}); \quad n_l \equiv \text{lines/image}, T_L \equiv \text{line time}, T_{LB} \equiv \text{time between lines}$$

$$\text{Max. time for frame processing in Sw: } T_{FPS} \leq T_F - T_T \quad (2)$$

The hardware blocks work within different clock domains to fit the special features of every subsystem. The *Video\_Domain* is synchronized with the camera’s clock. It uses to work at frequencies between 5 and 40 MHz. This clock manages the transfer of the video data coming from the camera’s interface block (*Camlface*), which uses to be composed of external circuits (video decoders for analog cameras or receivers/transceivers for digital cameras). These circuits usually have several configuration registers that can be written by the block *CamConf* with the information stored in the registers block *ConfRegs*. Video Data is decoded afterwards by the *VidInput* block that generates the video signal (with the appropriated format) and the information relative to horizontal/vertical synchronization.

The *Process\_Domain* is driven by the processing clock, it usually work at a higher frequency so as to speed up the data processing (up to 100 MHz or superior). No



specific *ImgProc* block is included in the reference architecture but several frameworks for the different types of processes. Following these frameworks the processing modules must be designed to form a pipeline by which data and synchronization or control signals flow jointly. The guidelines support segmentation at pixel or line levels, in order to start the work as soon as the pixel/lines arrive to the *ImgProc* block without waiting until the whole image is available. This reduces the necessary storage resources as well as the number of memory accesses, what can be a bottleneck in neighbourhood operations. In some cases, the whole image must be processed before to continue with the following task, e.g. connected component labelling algorithms, in these situations the segmentation will be made at image level. Once the data exits from this block, must be aligned and packaged for storing in the memory banks or for transmitting to the *SwIface* block. A parameterised *DataBlk* block was implemented to perform this task.

The *Interface\_Domain* is synchronized with the clock of the PCI local bus (up to 50 MHz). This domain includes the global control block, *GBCtrl*, and the configuration registers, *ConfRegs*, which store the configuration parameters for the different parts of the system. These blocks can be accessed directly by the Sw through the configuration bus, *ConfBus*, which is also used to send commands to the control block. The *TrigCtrl* block produces the trigger signal of the camera using the information collected by the sensors and following the parameters written on its internal configuration registers.

### 3 ConAVI a Codesign Environment for *On-Line* AVIs

Using as reference the *onAVI* architecture, a codesign/coverification environment has been developed. The environment has three basic components: an *onAVI* simulable and synthesizable design described with Handel-C, the framework for software side of the architecture, and a plugins library for cosimulation purposes.

The *onAVI* reference design was described with Handel-C. In order to provide the necessary flexibility to fit a wide range of applications, two strategies were adopted. Firstly, some libraries of blocks were built to cover different kind of devices. For example, a library of *CamIface* blocks for different video formats or a library of drivers for both asynchronous and synchronous static RAMs (SRAM, ZBT SSRAM).

Secondly, the code is parameterised to support the application particularities like image or pixel size. Some of the parameters are fixed in compilation time and others can be change in run time by means of software and *GBCtrl+ConfRegs* blocks.

Additionally, a common interface was defined to allow the interconnection of the image processing modules in a pipeline fashion. This is based on the *channels* mechanism that Handel-C provides. Communications over *channels* are performed seamlessly and can take from one up to four cycles, since those are implemented with a blocking scheme, which waits until sender and receiver become ready. The transfer protocol is very simple, every block in the pipeline has an associated latency during which is enable to accept data from the previous block. Once the internal pipeline is full, it has to wait for send new data until the next block in the pipe is prepared to accept it. Following this specification a library of image processing modules was coded. Its functionality is equivalent to software components usually found in commercial image processing libraries, reason why they are easily interchangeable facili-

tating the refinement process of the hw/sw partition. If others are necessary, the engineer can easily describe them on his own using the architecture framework provided.

For clock domains interconnection, *channels* for synchronization and double port memories for data transmission were chosen, although others structures as double clock FIFOs can be used since Handel-C can be combined with other HDLs.

The software framework includes the PCI drivers that depend on the reconfigurable board chosen (generally provided with it). In addition, a minimum staff of functions to send and receive data and commands are developed to meet the *SwIface* requirements. These functions are encapsulated in a Dynamic Link Library (dll) and can be called from any standard C code.

The Handel-C toolset, DK, provides several mechanisms for cosimulation. One of them is the *plugin*, informally speaking a *plugin* is a component described using software high-level languages (C/C++, Visual Basic, etc...) that can be connected to the Handel-C clock and ports, by means of a dll. This library (plugin.dll) incorporates the necessary functions to synchronize the component work with the system clock and to allow the data communications. This way the parts not included in the reconfigurable devices (cameras, electro-mechanicals components, software, etc...) were modelled using the *plugins* library, allowing the system can be verified under "realistic" conditions.

For example, a progressive scan camera were modelled including all the signals and functionalities of the real component: external trigger, line valid, frame valid, pixel clock, configuration bus, data bus, etc...The model generates the image data synchronously with the associated signals. The language employed was C/C++ that allows an easy access to image files to generate the test patterns for the system. A special *plugin* was developed to display partial results during the cosimulation. This *plugin*, called *viewer*, can be connected to any part of the processing pipeline to detect bugs in the image processing blocks. This viewer has a graphical interface that is firstly showed at the instance initiation. This interface allows showing the received information per pixels, per lines or per whole images.

Additionally, *HwIface* plugin was developed to emulate the functionality of the correspondent software subroutine. It has a simple semaphore like mechanism to access exclusively to the double buffer memory banks. It uses two ports, the State port (that is written by the Hw part) and the Control port (that is written by the Sw part) to synchronize the accesses. By means of these ports, it is possible to communicate messages as "data ready" or commands. It also has the buses that allow the system configuration.

Finally, using the *plugin* methodology the software part of the system was easily encapsulated and communicated with the hardware simulation. However, the data transfer timing cannot be modelled easily because it depends on indeterminist factors and not specified variables (e.g. the bus load). Thus, no timing model was adopted for software simulation, and consequently equation (2) must be verified on the final system.

The codesign flow defined by the ConAVI environment is stepwise refinement-based process. It starts from a functional description written in C/C++ that could come from a previous implementation of the AVIs using a commercial system. Then an initial version is built selecting the appropriated blocks in the *onAVI* architecture, and encapsulating the sw part on a *plugin*. The cosimulation is performed to check the

entire system functionality, the clock cycles and the resource usage (estimated by DK) of the hw side. In the other hand the execution time of sw side can be estimated with software profiling tools: if functionality requirements are not achieved, we should revise the hw design in the Handel-C code using its debugger tools; if speed requirements are not achieved, we can try to optimise the hw code to reduce the number of clock cycles or we can refine the partition moving new modules to hw.

Several iterations can be necessary until reach the specifications.

## 4 Results

A real problem of quality control of preserved orange segments was used as case study. The inspection must be performed while the segments travel on a conveyor belt and faulty segments have to be extracted by means of wind ejectors. To model the Electro-Mechanical subs. two *plugins* were added to the library; one for simulating an encoder that registers the conveyor belt movement and the other to simulating a row of ejectors. Basically, the Processing Subsystem is composed by three image processing blocks (two bias thresholding, 3x3 binary convolution and component labelling), one geometrical extractor, one statistical extractor (inertia central moments), and one classifier.

As implementation platform, the Nallatech Ballynuey and BallyVision boards were used. The first is a PCI board populated with one FPGA (XCV300) and two SSRAM banks, the second is an add-on board that include the video decoder, two banks of memory and a second FPGA.

Following the preliminary partition defined in the *onAVI*, the Acquisition, Electro-Mec. and Control Subsystems were mapped in hardware by means of library blocks. The initial implementation the platform gave a resource usage of 390CLBs(14%) + 8BRAM(50%) for the first FPGA and 576CLBs(18%) + 8BRAM(50%) for the second. The cosimulation of the *onAVI* without the *ImageProc* block takes only 59 seconds and represents the environment overhead in the cosimulation.

**Table 1.** Design space exploration with *ConAVI*

Sw//Hw partition	Hw (gates/RAMbits)	Throug. (img/s)	Cosim. (s)
Sw based system	0	5	--
I. Bin //Conv+Labell+ Geom+Stat+Class	99K /12.288	5,5	75
II. Bin+Conv // Labell+ Geom+Stat+Class	131K/13.738	9	181
III. Bin+Conv+Labell // Geom+Stat+Class	192K/54.730	18	212
IV. Bin+Conv+Labell+Geom // Stat+Class	200K/54.730	18	232

The *ConAVI* environment was used to explore the design space starting from a validated description written in C/C++ with the Matrox Image processing Libraries (MIL). Running this code on PentiumIII (@1GHz) workstation we get a performance of 5 images/sec (725x582). The *ImageProc* modules were gradually moved to hardware and the results are shown in the table below. First column represents the total hardware usage estimated by DK. The second column is the throughput of the system,

this was calculated based on the number of cycles obtained in the cosimulation and the  $T_T$  parameter measured on the real platform. The third column shows the time of cosimulation for the whole system, as can be seen the number of cycles per second simulated is enough to get tolerable times even for complex hw processing tasks.

The analysis of results allows the election of the best system implementation taking into account both, the performance requirements of the application and the physical resources of the platform.

## 5 Conclusions

This paper presents a codesign/coverification environment, *ConAVI*, especially conceived for the specific domain of the Automated Visual Inspection Systems. A codesign framework has been developed based on a preliminary partition derived from a generic architecture proposed in this work. This facilitates the system design and reduces the development time of hw/sw systems.

The tools chosen for the environment allow the simulation of the whole system and even the external hardware parts. The codesign flow starts with a software specification and both sides of the system are described in a high level language facilitating the movement of the blocks between hw and sw. It is shown that by using the proposed framework the exploration of the design space and the verification time of an AVI system can be similar to software development times.

## Acknowledgments

This work was supported by the Ministry of Education and Science of Spain under contract TIC2000-1765-C03.

## References

1. Baykut et al. Real-Time Defect Inspection of Textured Surfaces. *Real Time Imaging* 6, 17-27, 2000.
2. Radovan Stojanovic et al. Real-Time Vision-Bsed System for Textile Fabric Inspection. *Real-Time Imaging* 7, 507-518, 2001.
3. Batlle J. ; Martí, J.; ridao, P.; Amat, J. A new FPGA/DSP parallel architecture for real-time image processing. *Real-Time Imaging* 8, pp 345-35, 2002.
4. Zuloaga A. VHDL simulation tool for designing image processing systems. XIV Desing of Circuits and Integrated System Conference. Mallorca 1999.
5. Siavash Bayat S., Seyed Ghassem M., Ghazanfar Asadi. Fast Prototyping with Co-operation of Simulation and Emulation. *FPL2002, LNCS 2438*, pp.980-983, 2002.
6. Klaus Buchenrieder, Ulrich Nagendingerl, Andreas Pyttel, Alexander Sedlmeier. Integration of Reconfigurable Hardware into System-Level Design. *FPL2002, LNCS 2438*, pp. 987-996, 2002.
7. Miroslav Licko, Jan Schier, Milan Tichy, Markus Kühl. MATLAB/Simulink based Methodology for Rapid-FPGA-Prototyping. *FPL2003, LNCS2778*, pp. 984-987, 2003
8. <http://www.celoxica.com/>
9. Cuenca S., Ibarra F. et al. Reconfigurable Frame-Grabber for Real-Time Automated Visual Inspection Systems. *FPL2001, LNCS 2147*, pp. 223-231, 2001.

# Hardware-Accelerated Template Matching

Raúl Cabido, Antonio S. Montemayor, and Ángel Sánchez

Universidad Rey Juan Carlos, C/Tulipán, S/N  
28933 Móstoles, Madrid, Spain

{a.sanz,an.sanchez}@escet.urjc.es, rcabido@gmail.com

**Abstract.** In the last decade, consumer graphics cards have increased their power because of the computer games industry. These cards are now programmable and capable of processing huge amounts of data in a SIMD fashion. In this work, we propose an alternative implementation of a very intuitive and well known 2D template matching, where the most computationally expensive task is accomplished by the graphics hardware processor. This computation approach is not new, but in this work we resume the method step-by-step to better understand the underlying complexity. Experimental results show an extraordinary performance trade-off, even working with obsolete hardware.

## 1 Introduction

Object recognition problems are described as a labeling problem based on models of known objects [1]. Template matching is a very well known feature detection technique used in low level Image Processing and Computer Vision tasks, such as object recognition and tracking. As an image matching technique it compares portions of images against one another [2]. Many kind of implementations have been proposed, although the most basic one is related to the cross-correlation computation in order to compare the image and a pattern using a distance measure.

The template matching calculation involves a pixel by pixel analysis of the template into an image portion, evaluating every location of the target image. In a generalized approach template matching should be invariant under scale and rotation transformations. As a consequence, this technique is computationally very expensive.

Computer graphics have been very popular during the last two decades for the rapid expansion of computer generated special effects in films, multimedia and computer games. This fact has allowed the evolution of graphics hardware to unprecedented limits. Commodity graphics hardware has evolved since the mid 90's giving a considerable amount of programming power to developers in order to customize their rendering effects in real time. Apart from that, a consumer graphics processing unit (GPU) has become inexpensive and can be considered a kind of programmable stream processor. Their programmable capabilities has helped the development of applications far beyond rendering purposes. Many

authors have demonstrated that these consumer GPUs have a great raw performance, some times even superior to the most common and powerful CPUs [3–6]. They have been used as a co-processor for the central processing unit (CPU) remaining the idea that they can be encountered in most off-the-shelf desktop computers. Examples of it can be found in applications that exploits the power of the GPU for linear algebra calculations [9–12], physically-based simulations [6], image and volume processing [4, 13–15], neural network implementations [16, 17] or even acceleration of database operations [18] among others [19].

In this work we demonstrate that an efficient template matching based on cross-correlation calculation can be achieved using a commodity graphics card. It has been implemented exploiting the intrinsic parallelism of the graphics hardware. Additionally, we have proposed three kinds of models to increase the efficiency of the process step by step, showing details of their implementation and encountered difficulties. Apart from that, we propose a not-new but recent framework for image processing development that can give ideas to other researchers for customizing their implementations.

## 2 Graphics Hardware

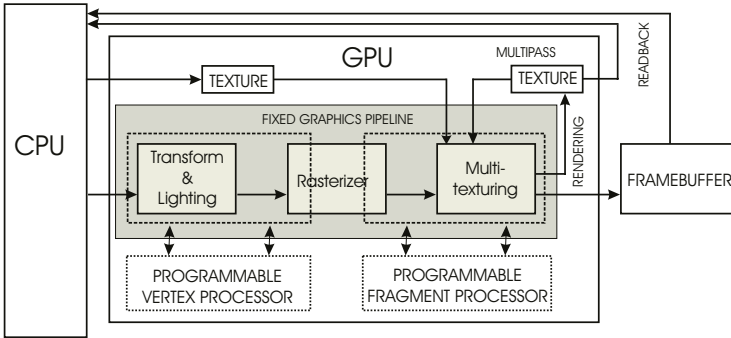
Commodity graphics hardware has evolved drastically since the mid 90's. With the aid of the rapid expansion of computer games and multimedia technologies these consumer graphics processing units (GPUs) have also become very powerful and inexpensive hardware.

Traditionally, these 3D graphics cards implemented a fixed pipeline for the processing of primitive descriptions tuned as a state machine from an API such as OpenGL. But their previously fixed graphics pipeline stages were replaced with programmable components, the transform and lighting (T&L) and the multi-texturing one, providing great versatility and power to the developer [7].

The hardware accelerated programmability of GPUs has been exposed to programmers for the development of programs called shaders. These shaders are loaded into the graphics card for replacing the fixed functionality. There are two kinds of shaders, respectively called vertex and fragment shaders. They constitute the executable code of the corresponding programmable components of the graphics pipeline. These shaders are primarily used for rendering complex special effects and realistic images in real-time. The basic CPU/GPU architecture model is outlined in Figure 1.

The programmability of the GPU is very well suited for stream computations, in which a simple kernel operation is executed over a large number of elements in a single-instruction multiple-data (SIMD) fashion [8, 9].

Textures and the multi-texturing capability provides some ways of efficient SIMD computation. In this context, a texture is an image that can be mapped to a polygonal structure to provide realism to the model. Basically, as an image, it can represent four values (R, G, B, A) as color and transparency components in every accesible location, called fragments or texels. The programmer is responsible for organizing its data in a grid way to convert them into a texture, so creating textures in which texels keep numerical values of interest. As it will be



**Fig. 1.** Basic CPU/GPU programming model. When enabled, programmable vertex and fragment execution paths replace their corresponding stages of the fixed graphics pipeline (represented in dot-lines). Also note the possibility of direct rendering to framebuffer or rendering to another texture (pbuffer), that can be used again as input data in the multipass approach.

shown in Section 3 and 4, it is desirable to fill the whole capacity of the textures. This is because, in the fragment program, the processing cost of a single channel in comparison to the processing cost of the entire quadruple (RGBA) is similar.

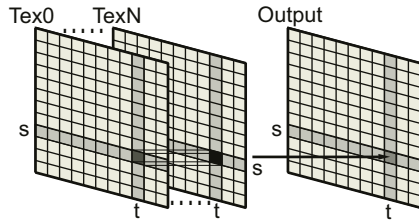
In order to operate on the texels, the texture is fixed to a well determined grid. Then, a custom fragment shader is enabled and the operation kernel is executed over every fragment by simply rendering. A schematic visualization of a number of textures applied as input for a fragment program can be seen in Figure 2.

The output result can be redirected to the input (by means of a pbuffer) in a multi-pass approach for continuing the processing task (see Figure 1). At this point, it is important to remark that the readback process from video memory to host memory after the rendering step is a computation bottleneck.

In this model, a shader is a program executed by the GPU. Originally they had to be coded in assembly, but as the graphics hardware increased in functionality and programmability, these shaders were more difficult to implement. Even more, the rapid evolution of GPUs forced to rewrite previous shaders to get maximum performance when a new family of graphics hardware were released. The solution came with the apparition of commercial high level shading languages and their compilers, which helped in portability and legibility, so improving efficiency in the development process. Nvidia's Cg, Microsoft's HLSL and recently OpenGL Shading Language have been the first commercially available languages for commodity graphics hardware with major acceptance. A brief classification, chronology and explanation of these languages can be found in [20].

### 3 Application to Pattern Recognition: Template Matching

We have developed three models of a basic template matching for being processed on the GPU. Each one of the models increase its complexity but also its efficiency. For demonstration purposes they do not consider rotations and scales.



**Fig. 2.** Simple fragment program computation. A common fragment program will be executed over every position of the input textures (*Tex0-TexN*), for example returning a value at  $(s,t)$  in the output texture for values at  $(s,t)$  in the input textures.

The first model does not exploit the SIMD capability of the graphics pipeline. It only computes the cross-correlation between the template and each location inside the image in a GPU architecture approach.

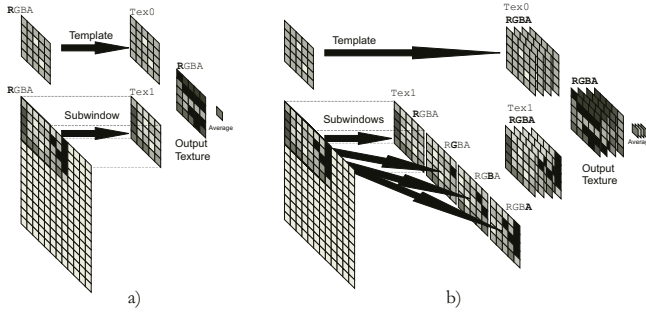
The second model uses the four channels (RGBA) of the template and target textures to compute the cross-correlation in four different positions at a time (at each rendering pass). This model is preferable in respect to the previous one although it is not as efficient as it could be.

The third model uses the same philosophy as the second one but computes much more positions. It exploits not only the RGBA data allocation fashion but also the repetition of the template in a texture of equal size than the target image texture.

For every model we consider three stages of processing: initialization, distance measurement calculation for a single position and new position estimation or updating. In the initialization stage we can preprocess the input images. In this application, if input images were in color format, an RGB to gray scale conversion is executed as a preliminar rendering with the appropriate fragment program enabled. Another kind of preprocessing could be done in this stage, such as a gaussian filtering to reduce noise artifacts.

An intrinsic limitation of the GPU architecture is the lack of global registers in the programmable rendering pipeline. For the template matching, a subwindow is computing a distance function among pixels and, after that, a sum for all distance values is computed in order to get the cross-correlation result. This sum has to be kept for every position to compare the evaluations. As there is no accumulative register, this sum of distance values must be computed in an alternative fashion. A typical way to proceed is by asking for a reduced level of detail (mipmap) of the output texture. That level will offer average values of the corresponding neighborhood of the texture at each fragment. With that average it is easy to obtain the corresponding summation multiplying by the ratio between the number of fragments of the original texture and the number of fragments of the resulted mipmap. This process can be done for power-of-two textures and in cases where precision of one byte is enough for the result. In other cases, a “ping-pong” strategy with two puffers is needed.





**Fig. 3.** a) Model 1 only takes advantage of the R channel and *Tex1* have to be updated at each evaluation to complete the matching. b) Model 2 takes advantage of the RGBA channels. Four subwindows are loaded into an RGBA texture (*Tex1*). Also, the gray scale template is repeated in each channel of *Tex0*. Thus, four positions are evaluated in each pass.

### 3.1 Model 1: Red Channel Exploitation

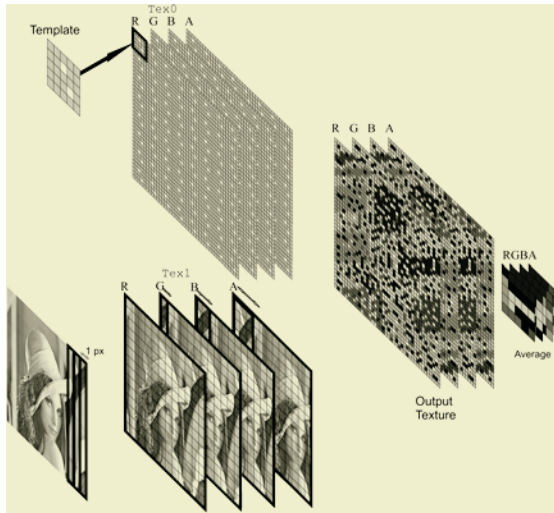
This is the most limited model but it is easy to understand. It does not exploit the entire quadruple of an RGBA texture, but it can open the mind for next approaches.

First, we load the template and an equal size portion of the target image into two textures (*Tex0* and *Tex1*). The RGBA textures have the gray value of each fragment in its RGB channels. It is easy to calculate the difference between the R channels of both textures fragment-by-fragment. By the explained limitation of the GPU, the global value of the matching fitness is done reducing the output texture to one pixel, thus returning in the pixel the average of the whole output texture. This average value is a proportional measure of the summation, and can be kept in system memory to be compared to other results. Figure 3.a outlines the procedure. The performance of this model is very limited because of in each pass we just evaluate one position. Once we have evaluated the matching of the template in that position, we have to “move” the texture coordinates of the target image to load another subwindow and repeat the entire process. Then, the number of rendering passes is proportional to the number of pixels of the target image.

### 3.2 Model 2: RGBA Channel Exploitation

This model exploits the RGBA texture channels. Now, the programmer is responsible for organizing its data to produce the textures. The key point is the decomposition of four portions of the target image into the RGBA channels of *Tex1* as shown in Figure 3.b. Also, it is important to note that the template is repeated in each channel of *Tex0* to make the fragment program work in RGBA.

Again, after each rendering pass the texture that contains the subwindows have to be updated (*Tex1*), allocating portions of the original image in the texture. This process can be computationally expensive, even a main bottleneck depending on the performance of the results readback.



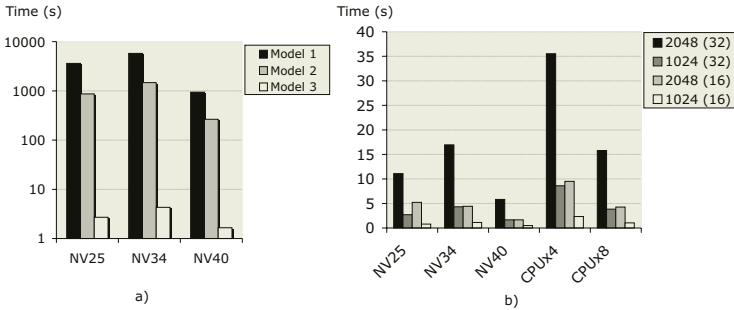
**Fig. 4.** In Model 3 the target image (Lenna) is loaded into the RGBA channels of *Tex1*, the arrows represent different offsets. The template image is loaded and repeated along the *Tex0*. In each rendering pass this model evaluates many positions in parallel.

### 3.3 Model 3: Vectorization and Repetition

This model is much more efficient than the previous ones. An outline of this model can be seen in Figure 4. The target image ( $N \times N$  pixels) is loaded into the RGBA channels of a texture (*Tex1*). However, there is an offset of 1 pixel in the horizontal direction in each channel except the first one. On the other hand the template ( $n \times n$  pixels) is loaded into the RGBA channels of *Tex0*, repeated until having a size equal to the target texture (*Tex1*). In this way, every texels will have correspondence one each other in the fragment program. The offset of the GBA channels of *Tex1* provides a way to evaluate 3 more different positions at the same time, so for each RGBA template subwindow of *Tex0*, 4 positions are evaluated. As the template is repeated  $\frac{N}{n} \times \frac{N}{n}$  times in *Tex0*, the number of parallel evaluations in each rendering pass is increased to  $4 \times \frac{N}{n} \times \frac{N}{n}$ . For a  $1024 \times 1024$  target image and a  $32 \times 32$  template this leads to 4096 evaluated positions for each rendering pass. The updating rule for this model is quite simple because it only needs the translation of the whole template texture over the target texture. This translation implies no data reallocation after the rendering pass and thus it eliminates a previous explained bottleneck.

## 4 Experimental Results

Experiments have been performed using 3 different graphics cards. The first platform is a Nvidia GeForce4 Ti4600 VGA card (NV25) in a 1.4GHz Pentium 4, 128 MB RAM, AGP $\times$ 4 (named CPU $\times$ 4). The second and third platforms use a 3.2GHz Pentium 4, 1GB RAM, AGP $\times$ 8 (named CPU $\times$ 8) and respectively a



**Fig. 5.** a) Different models for three different GPUs (target image: 1024x1024; template image: 32x32). b) Comparison between CPU vs GPU (Model 3) performance. Different sizes of square templates and target images are considered.

Nvidia GeForce FX5200 (NV34), and a Nvidia 6800GT (NV40). GPU applications have been coded in C using OpenGL as rendering API, Cg 1.2 as shading language and Nvidia v66.93 drivers, while CPU programs were coded in C.

Figure 5 shows the experimental results. We have considered the previous 3 models for the GPU implementation and 5 different platforms. In respect to the GPU implementations Model 1 is the worst in terms of efficiency, while, as described before, Model 3 exploits the intrinsic parallelism of the graphics card. Note that Figure 5 a) is semilogarithmic. Figure 5 b) shows that in the latest GPU the performance of this algorithm is very high, even superior to its host CPU performance.

## 5 Conclusions

We have presented a step-by-step alternative implementation of a well known object recognition method. We have demonstrated that the processor of a modern graphics card can afford better performance than a modern CPU under certain conditions, in particular, allocating data in a regular and parallel manner. The GPU should operate in a SIMD fashion to get the most performance hit. Experimental results show that the graphics card can be exploited in order to execute non-rendering tasks freeing some computational load to the CPU.

## References

1. Jain, R., Kasturi, R., Schunck, B. G.: Machine Vision. McGraw-Hill, Computer Science Series (1995).
2. Baxes, G. A.: Digital Image Processing, Principles and Applications. John Wiley & Sons, Inc (1994).
3. Thompson, C.J., Hahn, S., Oskin, M.: Using Modern Graphics Architectures for General-Purpose Computing: A Framework and Analysis, *Int. Symposium on Microarchitecture (MICRO), 2002*.

4. Goodnight, N., Wang, R, Woolley, C., Humphreys, G.; Interactive Time-Dependent Tone Mapping Using Programmable Graphics Hardware, *Eurographics Symposium on Rendering*, (2003) 1–13.
5. Purcell, T.: Ray Tracing on a Stream Processor, Ph. D Thesis, Univ. of Stanford (2004).
6. Harris, M. J.: Real-Time Cloud Simulation and Rendering, Ph. D Thesis, Univ. of North Carolina at Chapel Hill (2003).
7. Olano, M.: A Programmable Pipeline for Graphics Hardware. Ph.D. thesis, University of North Carolina at Chapel Hill (1998).
8. Venkatasubramanian, S.: The Graphics Card as a StreamComputer, *Workshop on Management and Processing of Data Streams, San Diego, California, USA* (2003).
9. McCool, M., Du Toit, S., Popa, T., Chan, B., Moule, K.: Shader Algebra, *ACM Transactions on Graphics* (2004).
10. Larsen, E. S., McAllister, D.: Fast Matrix Multiplies using Graphics Hardware, *In Proc. Supercomputing 2001*.
11. Bolz, J., Farmer, I., Grinspun, E., Schröder, P.: Sparse matrix solvers of the GPU: Conjugate gradients and multigrid, *ACM Trans. on Graphics*, (2003) 917–924.
12. Kruger J., Westermann R.: Linear algebra operators for GPU implementation of numerical algorithms. *ACM Trans. on Graphics* (2003) 908-916.
13. Yang, R., Welch, G.: Fast Image Segmentation and Smoothing Using Commodity Graphics Hardware, *Journal of Graphics Tools*, (2002) **7(4)**:91–100.
14. Colantoni, P., Boukala, N., da Rugna, J.: Fast and Accurate Color Image Processing Using 3D Graphics Cards, *Proc. of 8th Int. Workshop on Vision, Modeling and Visualization, Germany* (2003).
15. Krueger, J., Westermann, R.: Acceleration Techniques for GPU-based Volume Rendering. *In Proc. IEEE Visualization 2003*.
16. Bohn, C.A.: Kohonen Feature Mapping Through Graphics Hardware. *In Proc. of 3rd Int. Conference on Computational Intelligence and Neurosciences 1998*.
17. Oh K.-S. and Jung K.: GPU implementation of neural networks, *Pattern Recognition*, (2004) **37**: 1311–1314.
18. Govindaraju, N.K., Lloyd, B., Wang, W., Lin M.C., Manocha, D.: Fast Computation of Database Operations using Graphics Processors, *In Proc. SIGMOD 2004, Paris, France* (2004).
19. GPGPU Website, <http://www.gpgpu.org>
20. Rost, R.J.: OpenGL Shading Language. Pearson Education (2004).

# Author Index

- Aissani, Mohamed II-529  
Alba-Castro, José Luis II-513  
Alcántara, Enrique II-658  
Alemany, Sandra II-658  
Almeida, Daniel I-128  
Altamirano-Robles, Leopoldo II-720  
Álvarez, Nancy A. I-227  
Álvarez-Borrego, Josué II-83  
Álvarez Marañón, Gonzalo I-411  
Amores, Jaume I-28  
Ancona, Nicola I-277  
Andrés, Jesús II-622  
Ángel, Luis I-631  
Angulo, Jesús II-199  
Antequera, Teresa II-697  
Antón-Canalís, Luis I-553  
Appleboim, Eli II-405  
Araújo, Hélder I-168  
Armingol, José Maria I-579  
Avci, Engin I-594  
Ávila, Mar II-697
- Badenas, Jorge I-227  
Baldrich, Ramón I-192, II-666  
Bandeira, Lourenço II-387  
Barandela, Ricardo II-59  
Barceló, Lluís I-77  
Barreira, Noelia I-60  
Batista, Jorge P. I-200  
Baumela, Luis I-476  
Bellens, Rik I-545  
Benedí, José Miguel II-163, II-586  
Benhammadi, Farid II-529  
Bernardino, Alexandre I-11, I-335, I-537  
Bernués, Emiliano I-484  
Beucher, Serge I-119  
Bey-Beghdad, Kadda II-529  
Bez, Helmut E. I-384  
Biancardi, Alberto II-705  
Biancardini, Loic I-119  
Binefa, Xavier I-77  
Bioucas-Dias, José M. I-360  
Böhm, Matthias II-75  
Borràs, Agnés I-325  
Bosch, Anna II-311, II-431
- Bruno, Roberto II-674  
Buenapósada, José Miguel I-476  
Bui, Tien D. II-319  
Bunke, Horst II-99, II-147
- Cabido, Raúl I-691  
Caldas Pinto, João R. II-387  
Calera-Rubio, Jorge II-649  
Cámara, Antonio I-683  
Campilho, Aurélio C. II-191  
Cantador, Iván II-43  
Caro, Andres II-697  
Carreira, María J. II-175  
Carvalho, Paulo C.P. II-479  
Casacuberta, Francisco II-605, II-614, II-622  
Casañ, Gustavo I-227  
Castillo, Carlos I-209  
Castrillón-Santana, Modesto I-445, I-461, I-553  
Castro, María José I-376  
Chamorro-Martínez, Jesús I-52  
Chandra, Sunil II-183  
Chang, Carolina I-209  
Chang, Jae Sik I-176  
Chen, Wensheng II-67  
Cheng, Jian I-587  
Cheng, Lishui I-285  
Cheng, Yun I-419  
Chetverikov, Dmitry II-223  
Choung, Young Chul II-279  
Chung, Paul W.H. II-343  
Civera, Jorge II-630  
Coito, Fernando J. II-19  
Collado, Juan Manuel I-579  
Costeira, João P. I-102  
Coutinho, David Pereira II-355  
Cubel, Elsa II-630  
Cuenca, Sergio I-667, I-683  
Cuevas-Martínez, Juan C. II-571  
Cuoghi, Lorenza II-674  
Cyganeck, Bogusław I-308
- Dai, Daoqing II-67  
Dai, Haisheng II-555  
Dai, Kui I-419  
Darbon, Jérôme I-351

- de la Cruz, Jesús Manuel I-563  
de la Escalera, Arturo I-579  
De Witte, Valérie I-368  
del Rey, Ángel Martín I-411  
Delon, Julie II-239  
Denis, Nicolas I-401  
Déniz-Suárez, Oscar I-445  
Desolneux, Agnes II-239  
Deufemia, Vincenzo II-328  
D'Haeyer, Johan I-545  
Dias, José M.B. II-207  
Díaz-Ramírez, Víctor H. II-83  
Dickinson, Peter II-99  
Distante, Arcangelo I-277  
Dokladalova, Eva I-119  
Domínguez, Antonio I-217  
Domínguez, Sergio I-401  
Dorronsoro, José R. II-43  
Dosil, Raquel II-287  
du Buf, Hans I-128, II-255  
Durán, Marisa Luisa II-697
- Edirisinghe, Eran A. I-384, II-343  
España, Salvador I-376  
Espunya, Joan II-311, II-471
- Falcón-Martel, Antonio I-445  
Fan, Xian I-285  
Fang, Bin I-259, I-427, II-505  
Fazekas, Attila I-469  
Fdez-Vidal, Xosé R. II-287  
Fernández, César I-623  
Ferreira, Ricardo I-102  
Ferrer, Miquel II-139  
Figueiredo, Mário A.T. II-355  
Fisher, Mark I-292  
Flandrin, Georges II-199  
Fofi, David I-145  
Forest, Josep I-145  
Freixenet, Jordi II-431  
Fuertes, José M. I-20, I-251
- Gabarra, Elise II-371  
Galán-Perales, Elena I-52  
Gallardo, Ramiro II-697  
Gao, Song II-319  
García, Daniel I-110  
García, Miguel Ángel II-215  
García-Hernández, José II-658  
García-Perera, L. Paola II-579  
García-Sevilla, Pedro II-689
- García Varea, Ismael II-614  
Gattass, Marcelo II-479  
Gautama, Sidharta I-545  
Gedda, Magnus II-421  
Ghilardi, Paolo II-705  
Gil, Arturo I-623  
Gil, Pablo II-295  
Gómez, Pedro A. II-614  
Gómez-Ballester, Eva II-3  
González, Carolina I-631  
González, Jordi I-85, I-529  
González, Juan Carlos II-658  
González-Fraga, J. Ángel II-83  
González-Jiménez, Daniel II-513  
Gracia-Roche, Juan José I-484  
Granlund, Gösta I-3  
Gruber, Peter II-75  
Guerra, Cayetano I-184, I-217  
Guerrero, José J. I-69  
Guo, Jianjun I-419  
Guo, Qiang II-487  
Guzman, Giovanni I-316
- Hancock, Edwin R. I-235, I-268, II-123,  
II-155, II-247  
Hao, Yu II-397  
Haxhimusa, Yll II-107  
He, Zhenyu I-259  
Hentous, Hamid II-529  
Heras, Stella II-658  
Hernández, Daniel I-184, I-217  
Hernández, Mario I-217  
Hernández Encinas, Luis I-411  
Hernández-Sosa, Daniel I-461  
Herrero, José Elías I-484  
Hidalgo, José Luis I-376  
Hilario, Cristina I-579  
Hong, Helen II-463  
Huang, Jian II-67, II-505  
Huang, Xiangsheng I-453, I-492, II-51
- Ichikawa, Makoto I-137  
Iñesta, José M. II-649  
Isern-González, José I-184, I-445
- Jang, Sang Su I-607  
Jin, Biao II-447  
Juan, Alfons II-363, II-622, II-630, II-658  
Jung, Do Joon I-500
- Kang, Sung Kwan II-279  
Katsaggelos, Aggelos K. I-343, II-455

- Kerre, Etienne E. I-368  
Keyzers, Daniel I-511  
Kieś, Paweł II-336  
Kim, Cheong-Ghil I-675  
Kim, Eun Yi I-607  
Kim, Hang Joon I-176, I-500, I-607  
Kim, Sang Ho I-176  
Kim, Shin-Dug I-675  
Klaren, Arnoud C. I-36  
Klossa, Jacques II-199  
Kober, Vitaly II-83  
Kraetzl, Miro II-99  
Kraiss, Karl-Friedrich I-520  
Kropatsch, Walter G. II-107
- Lai, Ka H. II-343  
Lang, Elmar W. II-75  
Lapedriza, Àgata II-537  
Laurence, Pascal II-674  
Le Saux, Bertrand II-147  
Lee, Chang Woo I-500  
Lee, Jeongjin II-463  
Lee, Kisung I-392  
Lee, Su-Jin I-675  
Lemos, João M. II-19  
Letellier, Laurent I-119  
Levachkine, Serguei I-316  
Li, Jin Tao II-521  
Li, Ming II-397  
Li, Peihua I-161  
Linares, Diego II-586  
Lisani, Jose Luis II-239  
Liu, Jun I-453  
Liu, Ming II-447  
Liu, Wenbing II-91  
Liu, Yuncai I-647  
Lladós, Josep I-325, II-115  
Llobet, Rafael II-495  
Llorens, David I-571  
Lloret, Jaime II-605  
López, Antonio II-455  
López, Fernando II-666  
López Coronado, Juan I-110  
López-Ferreras, Francisco II-571  
López-Gutiérrez, Luis II-720  
López-Nicolás, Gonzalo I-69  
Loke, Eddy I-128  
Lorenzo-Navarro, Javier I-184, I-445, I-461  
Lu, Peng I-492  
Lucena, Manuel J. I-20, I-251
- Luengo-Oroz, Miguel A. II-199  
Lumini, Alessandra II-231, II-413  
Luo, Yupin II-555
- Malumbres, Manuel P. I-435  
Mariño, Cástor II-175  
Martí, Joan II-311, II-471  
Martínez, Pablo I-110  
Martínez-Baena, Javier I-52  
Martínez-del-Rincón, Jesús I-300  
Martínez-Usó, Adolfo II-689  
Martins, Nuno I-168  
Marzal, Andrés I-571  
Mas, Joan II-115  
Masip, David II-537  
Mata, Susana II-421  
Matabosch, Carles I-145  
Mazzeo, Pier Luigi I-277  
Mex-Perera, Carlos II-579  
Micó, Luisa II-3  
Miike, Hidetoshi I-137  
Ming, Xing II-545  
Moe, Anders I-44  
Molina, Rafael I-343, II-455  
Mollineda, Ramón A. II-27  
Monteiro, Fernando C. II-191  
Montemayor, Antonio S. I-691  
Montoliu, Raúl I-36  
Moreno, Marco I-316  
Moreno, Plinio I-11  
Moreno-Noguer, Francesc I-93  
Mouaddib, El Mustapha I-153  
Muñoz, Enrique I-476  
Muñoz, Xavier II-311
- Nácher, Beatriz II-658  
Nachtegaele, Mike I-368  
Nanni, Loris II-231, II-413  
Nascimento, José M.P. II-207  
Navarro, José R. II-622  
Nevado, Francisco II-614  
Ney, Hermann I-511  
Nolazco-Flores, Juan A. II-579, II-595  
Nomura, Atsushi I-137  
Nordberg, Klas I-3  
Nunes, Samuel I-128
- Oliveira, Paulo I-615  
Oliver, Arnau II-431, II-471  
Oliver, Jose I-435  
Oncina, Jose II-3

- Orghidan, Radu I-153  
 Orrite, Carlos I-484  
 Orrite-Uruñuela, Carlos I-300  
 Ortiz, Daniel II-614  
 Ortiz, Francisco II-295  
  
 Pagliardi, Matteo II-705  
 Paiva, Anselmo C. II-479  
 Pajares, Gonzalo I-563  
 Palazón, Vicente I-571  
 Pardo, Xosé M. I-60, II-287  
 Paredes, Roberto II-495, II-658  
 Pari, Lizardo I-631  
 Park, Hye Sun I-607  
 Park, Jong An II-279  
 Park, Min Ho I-607  
 Park, Se Hyun I-607  
 Pastor, Moisés II-363  
 Payá, Luis I-623  
 Peña-Díaz, Marco II-595  
 Pérez, José Alberto I-376  
 Pérez-Cortés, Juan C. II-495  
 Pérez de la Blanca, Nicolás I-20, I-251  
 Pérez-Sancho, Carlos II-649  
 Péteri, Renaud II-223  
 Penas, Marta II-175  
 Penedo, Manuel G. I-60, II-175  
 Peracaula, Marta II-471  
 Petro, Ana Belen II-239  
 Petrou, Maria II-183  
 Pina, Pedro II-387  
 Pinzolas, Miguel I-110  
 Piroddi, Roberta II-183  
 Pla, Filiberto I-36, I-227, I-537, II-35, II-689  
 Poyraz, Mustafa I-594  
 Prados-Suárez, Beén I-52  
 Puig, Domènec II-215  
 Pujol, Oriol II-11  
 Puntonet, Carlos G. II-75  
 Pylvänäinen, Timo I-639  
  
 Quintero, Rolando I-316  
  
 Raba, David II-431, II-471  
 Radeva, Petia I-28, II-11  
 Ragheb, Hossein II-247  
 Reinoso, Óscar I-623  
 Ribadas, Francisco Jose II-638  
 Ribeiro, Pedro I-537  
 Risi, Michele II-328  
 Rius, Ignasi I-85, I-529  
  
 Robison Fernlund, Joanne Mae II-713  
 Roca, Xavier I-85, I-529  
 Rodrigues, João II-255  
 Rodríguez, Luis Javier II-562  
 Rodríguez-Domínguez, Yeray I-461  
 Rogez, Grégory I-300  
 Rosa-Zurera, Manuel II-571  
 Rouco, José I-60  
 Rowe, Daniel I-85, I-529  
 Ruiz-Reyes, Nicolás II-571  
 Ruz, José Jaime I-563  
  
 Sagüés, Carlos I-69  
 Salgado-Garza, Luis R. II-595  
 Salvi, Joaquim I-145, I-153  
 San Pedro, José I-401  
 Sánchez, Ángel I-691  
 Sánchez, Gemma II-115  
 Sánchez, J. Salvador II-27, II-35, II-59  
 Sánchez, Joan Andreu II-163, II-586  
 Sánchez-Nielsen, Elena I-553  
 Sánchez-Palma, Pedro I-659  
 Sanchiz, José M. I-227  
 Sanfeliu, Alberto I-93, II-131, II-139, II-263  
 Santa, István I-469  
 Santos, João A. I-102  
 Santos-Victor, José I-11, I-335, I-537  
 Šára, Radim II-439  
 Saucan, Emil II-405  
 Schulte, Stefan I-368  
 Sebastián, Jose M. I-631  
 Sebe, Nicu I-28  
 Selmaoui, Nazha II-303  
 Serratos, Francesc II-131, II-139  
 Shang, Yanfeng II-447  
 Shet, Rupesh N. II-343  
 Shi, Fanhuai I-647  
 Shin, Yeong Gil II-463  
 Sigelle, Marc I-351  
 Silva, Aristófanos C. II-479  
 Silveira, Margarida II-271  
 Sintorn, Ida-Maria II-421  
 Smith, William A.P. I-268  
 Smutek, Daniel II-439  
 Sobral, João Luís II-682  
 Söderberg, Robert I-3  
 Sotoca, José M. II-27  
 Sousa, João M.C. II-387  
 Stadlthanner, Kurt II-75  
 Stella, Ettore I-277



- Strand, Robin I-243  
Suardíaz, Juan I-667, I-683  
Sun, Kun II-487  
Švec, Martin II-439  
Svensson, Stina II-421
- Tabbone, Antoine II-371  
Tabora Duarte, Margarida II-713  
Tang, Sheng II-521  
Tang, Yuan Yan I-259, I-427, II-505  
Teixeira, Ana R. II-75  
Theis, Fabian J. II-75  
Toledo, Ana I-659, I-667, I-683  
Tomás, Jesús II-605  
Tomé, Ana M. II-75  
Torres, Fernando II-295  
Torres, M. Inés II-562  
Torres, Miguel I-316  
Toselli, Alejandro Héctor II-363  
Tous, Francesc I-192  
Turkoglu, Ibrahim I-594
- Valadão, Gonçalo I-360  
Valdovinos, Rosa M. II-59  
Valiente, José Miguel II-666  
Van der Weken, Dietrich I-368  
Vanrell, María I-192, II-666  
Vázquez, Fernando II-35  
Vega, Miguel I-343  
Vera-Candeas, Pedro II-571  
Vergés-Llahí, Jaume II-263  
Vicente, Asunción I-623  
Vicente-Chicote, Cristina I-659, I-667  
Vidal, Enrique II-363, II-630  
Viksten, Fredrik I-44  
Vilar, Juan M. I-571  
Vilares, Jesus II-638
- Vilares, Manuel II-638  
Villanueva, Juan J. I-85  
Vitrià, Jordi II-537
- Wang, Haijing I-161  
Wang, Hongfang I-235  
Wang, Jianhua I-647  
Wang, Yangsheng I-453, I-492, II-51  
Wang, Zhengxuan II-545  
Wang, Zhiying I-419  
Wei, Xiaopeng II-91  
Wickramanayake, Dhammike S. I-384
- Xiao, Bai II-123  
Xu, Jin II-91
- Yang, Jianwei I-427  
Yang, Jie I-285, I-587  
Yang, Shiyuan II-555  
Yang, Xin II-447, II-487  
Yim, Yeni II-463  
You, Xinge I-259, I-427, II-505  
Yu, Hang II-155  
Yuen, Pongchi II-67
- Zahedi, Morteza I-511  
Zhang, Qiang II-91  
Zhang, Qing H. II-319  
Zhang, Xian II-379  
Zhang, Yong Dong II-521  
Zhou, Yue I-587  
Zhu, Ming II-447, II-487  
Zhu, Xiaodong II-545  
Zhu, Xiaoyan II-379, II-397, II-555  
Zhu, Xinshan I-492  
Zhu, Yanong I-292  
Zieren, Jörg I-520  
Zwiggelaar, Reyer I-292, II-431